# If I were to build a new operating system, what kind of features would it have? [closed]

Asked 16 years, 3 months ago Modified 12 years, 7 months ago

Viewed 12k times

30

As it currently stands, this question is not a good fit for our Q&A format. We expect answers to be supported by facts, references, or expertise, but this question will likely solicit debate, arguments, polling, or extended discussion. If you feel that this question can be improved and possibly reopened, visit the help center for guidance.

Closed 12 years ago.

I am toying with the idea of creating an completely new operating system and would like to hear what everyone on this forums take is on that? First is it too late are the big boys so entrenched in our lives that we will never be able to switch (wow - what a terrible thought...). But if this is not the case, what should a operating system do for you? What features are the most important? Should all the components be separate installations (in other words - should the base OS really have no user functionality and that gets added on by creating "plug-ins" kind of like a good flexible tool?)

Why do I want to do this... I am more curious about whether there is a demand and I am wondering, since the OSes we use most today (Linux, Windows, Mac OS X (Free BSD)) were actually written more than 20 years ago (and I am being generous - I mean dual and quad cores did not exist back then, buses were much slower, hardware was much more expensive, etc,...), I was just curious with the new technology if we would do anything differently?

I am anxious to read your comments.

**operating-system**

Share

Improve this question

Follow

edited Apr 28, 2012 at 18:50

Oded

**498k** ● 102 ● 893 ● 1k

asked Sep 19, 2008 at 11:57

Autobyte

**1,229** ● 4 ● 19 ● 28

1 cool question. why are all the cool questions not constructive? :-/ – HorseloverFat Aug 10, 2013 at 6:06

## 16 Answers

Sorted by: Highest score (default)

▲

To answer the first question: It's never too late. Especially when it comes to niche market segments and stuff like

23

that.

Second though, before you start down the path of creating a new OS, you should understand the kind of undertaking it is: it'd be a massive project.

Is it just a normal programmer "scratch the itch" kind of project? If so, then by all means go ahead -- you might learn alot of things by doing it. But if you're doing it for the resulting product, then you shouldn't start down that path until you've looked at all the current OSes under development (there are alot more than you'd think at first) and figured out what you'd like to change in them.

Quite possibly the effort would be better spent improving/changing an existing open source system. Even for your own experimentation, it may be easier to get the results you want if you start out with something already in development.

Share Improve this answer Follow

answered Sep 19, 2008 at 12:02

slicedlime
2,152 ● 1 ● 17 ● 16

3 Actually, it's not that bad. Building a multi-billion dollar OS, sure that is a massive undertaking. However, building an experimental OS is really not that bad. Minix 3 is less than 10,000 lines. – BobbyShaftoe Jan 10, 2009 at 3:09

I once attended a lecture called "building an OS in 2 hours". The lecturer did build large parts of an OS during this time, while explaining each step. Of course, this OS was barely bootable, but still. (Full disclosure: he didn't start with 0 lines of code, he had some basic, but still small, components ready). – keyser Apr 7, 2015 at 9:29

20

First, a little story. In 1992, during the very first Win32 ( what would become the MS Professional Developers Conference ) conference, I had the opportunity to sit with over some lunch with one Mr. Dave Cutler ( Chief Architect of what most folks would now know as Windows NT,Windows 2000, XP, etc. ).

I was at the time working on the Multimedia group at IBM Boca Raton on what some of you might remember, OS/2. Having worked on OS/2 for several years, and recognizing "the writing on the wall" of where OSes were going, I asked him, "Dave, is Windows NT going to take us into the next century or are there other ideas on your mind ?". His answer to me was as follows:

"M...., Windows NT is the last operating system anyone will ever develop from scratch !". Then he looked over at me, took a sip of his beer, and said, "Then again, you could wake up next Saturday after a particularly good night out with your girl, and have a whole new approach for an operating system, that'll put this to shame."

Putting that conversation into context, and given the fact I'm back in college pursuing my Master's degree ( specializing in Operating Systems design ), I'd say there's TONS of room for new operating systems. The thing is to put things into perspective. What are your target goals for this operating system ? What problem space is it attempting to service ?

Putting this all into perspective will give you an indication of whether you're really setting your sights on an achievable goal.

That all being said, I second an earlier commenters note about looking into things like "Singularity" ( the focus of a talk I gave this past spring in one of my classes .... ), or if you really want to "sink your teeth into" an OS in its infancy....look at "ReactOS".

Then again, WebOSes, like gOS, and the like, are probably where we're headed over the next decade or so. Or then again, someone particularly bright could wake up after a particularly fruitful evening with their lady or guy friend, and have the "next big idea" in operating systems.

Share Improve this answer Follow

edited Jun 2, 2009 at 7:29
John Topley
115k ● 47 ● 199 ● 240

answered Sep 19, 2008 at 13:39
user15749
221 ● 1 ● 2

+1 for Dave Cutler! Also, yeah I agree Singularity is interesting but I think maybe it gets too much attention.
– BobbyShaftoe Jan 10, 2009 at 3:15

"M...." this wouldn't be "marcelo.lopezjr" would it? lol
– Dustin Getz Jul 29, 2009 at 21:51

+1 for 'fruitful evening with their lady or guy friend' (although - if you're thinking about operating systems when you're having fruitful evenings with your lady or guy friends, you might be doing it wrong) – HorseloverFat Aug 10, 2013 at 6:09

15

Why build the OS directly on a physical machine? You'll just be mucking around in assembly language ;). Sure, that's fun, but why not tackle an OS for a VM?

Say an OS that runs on the Java/.NET/Parrot (you name it) VM, that can easily be passed around over the net and can run a bunch of software.

What would it include?

- Some way to store data (traditional FS won't cut it)
- A model for processes / threads (or just hijack the stuff provided by the VM?)
- Tools for interacting with these processes etc.

So, build a simple Platform that can be executed on a widely used virtual machine. Put in some cool functionality for a specific niche (cloud computing?). Go!

Share Improve this answer Follow

answered Sep 19, 2008 at 12:11

Daren Thomas
70.2k ● 42 ● 155 ● 205

1 I'm not sure this even makes sense. – BobbyShaftoe Jan 10, 2009 at 3:10

4 Actually, I think it might just make sense: Think of an application as a machine in a specific "state". If you can define that machine (the OS) and how it works, then the application can be scaled here and there without worrying about what its actually running on... – Daren Thomas Jan 19, 2009 at 12:41

1 A virtual machine is not a platform for an operating system, it's a sandbox within an *existing* OS... an "Operating System" running in the .NET VM doesn't make any sense... – Dave Swersky Jan 29, 2009 at 16:46

2 Actually, Dave, I believe this is *just* what the clouds are: Operating Systems that abstract the physical machine a lot more radical than Win32 and that crowd. AppEngine is an OS with IO, API etc. Who cares what OSes are running those services? – Daren Thomas Jan 30, 2009 at 16:30

I believe that instead you can virtualize a processor and then write an OS for that processor - this is what VMWare and Virtual PC does? But this is the same a writing for bare metal no? – Autobyte Jul 31, 2009 at 20:10

▲

7

▼

For more information on the micro- versus monolithic kernel, look up Linus' 'discussion' with Andrew Tanenbaum.

Share Improve this answer Follow

answered Sep 19, 2008 at 11:58

Roel
19.6k ● 7 ● 64 ● 97

▲

5

▼

I would highly suggest looking at an early version on linux(0.01) to at least get your feet wet. You're going to mucking about with assembly and very obscure low-level stuff to even get started (especially getting into protected mode, multi-tasking, etc). And yes, it's probably true that the "big boys" already have the market cornered. I'm not telling you NOT to do it, but maybe doing some work on the linux kernel would be a better stepping stone.

Share Improve this answer

Follow

answered Sep 19, 2008 at 12:01

basszero

30k ● 9 ● 58 ● 80

---

▲

5

▼

Check out Cosmos and Singularity, these represent what I want from a futuristic operating system ;-) Edit : SharpOS is another managed OS effort. Suggested by yshuditelu

Share Improve this answer

Follow

edited May 23, 2017 at 11:46

community wiki

8 revs, 2 users 90%

Pop Catalin

Cosmos (-t), also SharpOS is another C# operating system in development. In addition, they are attempting to use only managed code. – Timothy Carter Sep 19, 2008 at 12:24

Typo fixed, thank you also for the SharpOS reference, didn't knew about it. – Pop Catalin Sep 19, 2008 at 14:00

I think Singularity is neat and a great project but I'm not convinced that is the future of operating systems. – BobbyShaftoe Jan 10, 2009 at 3:10

@BobbyShaftoe - it has to be, it's a matter of economics, think how much money are spent today to make sure the OS has no vulnerabilities like buffer overflows, how many processes are involved in doing this and how much is spent for security fixes ... – Pop Catalin Jan 12, 2009 at 23:05

...managed code simply avoids all overflow underflow errors from start, and as Singularity has shown it can also have the required performance. I think it is a matter of time until a managed OS enters the market. – Pop Catalin Jan 12, 2009 at 23:07

4

An OS should have no user functionality at all. User functionality should be added by separate projects, which does not at all mean that the projects should not work together!

If you are interested in user functionality maybe you should look into participating in existing Desktop Environment projects such as GNOME, KDE or something.

If you are interested in kernel-level functionality, either try hacking on a BSD derivate or on Linux, or try creating your own system -- but don't think too much about the user functionality then. Getting the core of an operating system right is hard and will take a long time -- wanting to reinvent everything does not make much sense and will get you nowhere.

Share Improve this answer

Follow

answered Sep 19, 2008 at 12:02

Matthias Kestenholz

**3,348** ● 1 ● 23 ● 26

I think this misses the point of the question. Anyway, as to your first paragraph, why should an OS not have any "user functionality." There are some pretty radical ideas popping up in OS research nowadays. – BobbyShaftoe Jan 10, 2009 at 3:07

2

You might want to join an existing OS implementation project first, or at least look at what other people have implemented.

For example AROS has been some 10 or more years in the making as a hobby OS, and is now quite usable in many ways.

Or how about something more niche? Check out Symbios, which is a fully multitasking desktop (in the style of Windows) operating system - for 4MHz Z80 CPUs (Amstrad CPC, MSX). Maybe you would want to write something like this, which is far less of a bite than a full next-generation operating system.

Share Improve this answer Follow

answered Sep 19, 2008 at 12:29

JeeBee
17.5k ● 5 ● 52 ● 60

---

2

Bottom line...focus on your goals and even more importantly the goals of others...help to meet those needs. Never start with just technology.

I'd recommend against creating your own Operating System. (My own geeky interruption...Look into Cloud Computing and Amazon EC2)

I totally agree that it would first help by defining what your goals are. I am a big fan of User Experiences and thinking of not only your own goals but the goals of your audience/users/others. Once you have those goals, then move to the next step of how to meet it.

Now days what is an Operation System any way? kernal, Operating System, Virtual Server Instance, Linux, Windows Server, Windows Home, Ubuntu, AIX, zSeries OS/390, et al. I guess this is a good definition of OS... Wikipedia

I like Sun's slogan "the Network is the computer" also...but their company has really fallen in the past decade.

On that note of the Network is the computer... again, I highly recommend, checking out Amazon EC2 and more generally cloud computing.

Share Improve this answer Follow

edited Sep 19, 2008 at 12:33

answered Sep 19, 2008 at 12:28

user16651

197 ● 2 ● 6

But "cloud computing" is a fundamentally different issue. – BobbyShaftoe Jan 10, 2009 at 3:12

2

I think that building a new OS from scratch to resemble the current OSes on the market is a waste of time. Instead, you should think about what Operating System will be like 10-20 years from now. My intuition is that they will be so different as to render them mostly unrecognizable by today's standards. Think of frameworks such as Facebook (gasp!) for models of how future OSes will operate.

Share Improve this answer Follow

answered Jan 29, 2009 at 16:29

aduric

---

1

I think you're right about our current operating systems being old. Someone said that **all operating systems suck.** And yes, don't we have problems with them? Call it BSOD, Sad Mac or a Kernel Panic. Our filesystems fail, there are **security and reliability** problems.

Microsoft pursued interesting approach with its Singularity kernel. It isolates processes in software, using a virtual machine similar to .NET, and **formal verification** methods. Basically all IPC seems to be formally specified and verified, even before a program is ran.

But there's another problem with it - Singularity is **only a kernel.** You can't run application not designed for it on it. This is a huge penalty, making eventual transition (Singularity is not public) quite hard. If you manage to produce something of similar technical advantages, but with a real transition plan (think about IPv4->IPv6 problems, or how Windows got so much market share on desktop), that could be huge!

But **starting small** is not a bad choice either. Linux started just like this, and there are many cases when it leads to better design. Small is beautiful. Easier to change. Easier to grow. Anyway, good luck!

Share Improve this answer

Follow

answered Sep 19, 2008 at 13:17

Paweł Hajdan

**18.5k** ● 9 ● 52 ● 65

---

1

checkout singularity project, do something revolutionary

Share Improve this answer

Follow

answered Sep 19, 2008 at 13:40

Abhishek Mishra

**5,030** ● 8 ● 38 ● 38

---

1

I've always wanted an operating system that was basically nothing but a fresh slate. It would have built in plugin support which allow you to build the user interface, applications, whatever you want.

This system would work much like a Lua sandbox to a game would work, minus the limitations. You could build a plugin or module system that would have access to a variety of subsystems that you would use. For example, if you were to write a web browser application, you would need to load the networking library and use that within your plugin script. Need 'security' ? Load the library.

The difference between this and Linux is that, Linux is an operating system but has a windows manager that runs over top of it. In this theoretical operating system, you would be able to implement the generic "look" and "feel" of a variety of windows within the plugin system, or could you create a custom interface.

The difference between this and Windows is that its fully customizable, and by fully I mean if you wanted to not implement any cryptography at all, you can do that, or if you wanted to customize an already existing window, you can do that. Nothing is closed to you.

In this theoretical operating system, there is an OS with a plugin system. The plugin system uses a simple and powerful language.

Share Improve this answer Follow

answered Jan 29, 2009 at 15:59

user29053

---

1

If you're asking what I'd like to see in an operating system, I can give you a list. I am just getting into programming so I'm not sure if any of this is possible, but I can give you my ideas.

1. I'd like to see a developed operating system (besides the main ones) in which it ISN'T a pain to get the

wireless card to work. That is my #1 pet peeve with most of the ones I've tried out.

2. It would be cool to see an operating system designed by a programmer for other programmers. Have it so you can run programs for all different operating systems. I don't know if that's possible without having a copy of windows and OSX but it would be really damn cool if I could check the compatablity of programs I write with all operating systems.

Share Improve this answer Follow

answered May 16, 2011 at 20:46

NnyValentine
11 ● 1

---

▲
0
▼

You could also consider going with MINIX which is a good starting point.

Share Improve this answer Follow

answered Jun 2, 2009 at 7:20

Ankur Chauhan
1,390 ● 2 ● 18 ● 39

---

▲
0
▼

To the originator of this forum, my hats off to you sir for daring to think in much bolder and idealistic terms regarding the IT industry. First and foremost, Your questions are precisely the kind you would think should engage a much broader audience given the flourishing Computer Sciences all over the globe & the openness taught to us by the Revolutionary Linux OS, which has only begun to win the hearts and minds of so many out there by way of strengthing its user-friendly interface. So kudos on pushing the envelope.

If I'm following correctly, you are supposing that given the fruits of our labor thus far, the development of further hardware & Software concoctions could or at least should be less conventional. The implication, of course, is that any new development would reach its goal faster than what is typical. The prospect, however, of an entirely new OS system @this time would be challenging - to say the least - only because there is so much friction out there already between Linux & Windows. It is really a battle between open source & the proprietary ideologies. Bart Roozendaal in a comment above proves my point nicely. Forget the idea of innovation and whatever possibilities may come from a much more contemporary based Operating System, for such things are secondary. What he is asking essentially is, are you going to be on the side of profit or no? He gives his position away easily here. As you know, Windows is notorious for its monopolistic approach regarding new markets, software, and other technology. It has maintained a deathgrip on its hegemony since its existence and sadly the windows os is racked with endless bugs & backdoors.

Again, I applaud you for your taking a road less travelled and hopefully forgeing ahead and not becoming discouraged. Personally, I'd like to see another OS out there...one much more contemporary.

Share Improve this answer Follow

answered Jul 29, 2009 at 21:36

Tarus