



systemE

A lightweight systemd replacement written in Emacs lisp

What people are saying

“Kind of taking that whole “Emacs is an operating system” thing a bit too far, aren’t we?”

“At last we have solved the init controversy.”

“Well I guess while you’re writing this you aren’t creating atomic robots to take over the earth.”

To quote Strong Bad, this is impressive, disturbing, and makes me uncomfortable!

So close to GNU/Emacs/Linux

I think this is really cool, but calling it a “systemd replacement” feels like click bait to me.

It’s called a joke, guys. The README is literally full of them.

systemd and its ecosystem isn’t built inside of Emacs. > That’s because systemd is a mild editor lacking a decent lisp os

About

Using the tooling in this repo, I am able to boot from linux to sinit as PID1, and from there to Emacs acting as PID2 using `-script` mode, performing all typical `rc.boot` system initialization using Emacs lisp until we hit the getty.

```
Decompressing Linux... Parsing ELF... Perform
Booting the kernel.
Begin systemE initialization...Linux 5.4.0

Mounting proc...
(0 )
Mounting sys...
(0 )
Mounting run...
(0 )
Mounting dev...
(1 mount: mount: dev: Resource busy
)
Creating /run directories...
Mounting devpts...
(0 )
Mounting shm...
(0 )
Starting eudev...
(0 )
(0 )
(0 )
(0 )
Remounting rootfs as ro...
(0 )
Checking filesystems...
Remounting rootfs as rw...
(0 )
Mounting all local filesystems...
(0 )
Seeding random...
Setting up loopback...
(0 )
Setting hostname...
(Shell command succeeded with no output)
0
Exiting eudev...
(0 )
Storing dmccg output to /var/log...
Starting process supervisor...
Boot stage complete...
kiss login:
```

Status

- The rc.boot.el and rc.shutdown.el scripts are finished and I use them on my local machine. No support for fancy stuff like luks is planned.

-
- We still depend on suckless “sinit” for PID1 (Emacs is PID2) and this needs to be rewritten in a lisp. While **it is easy to execute Emacs as PID 1**, as evidenced here, Emacs does not reap zombies and I do not know how to make it reap them yet. I have found Emacs listens for the URS1 and URS2 signals, but not SIGCHLD which is what we need. We also need to figure out how to call wait () after that.
 - We still depend on busybox runit for a process supervisor and this needs to be reimplemented in elisp, or atleast converted to gnu shepherd. An Elisp service supervisor that could be communicated to using emacsclient as root would be really fun.
 - One pain point is getting a statically compiled Emacs. We don’t require this, but it sure would be nice. Unfortunately, when I statically compile Emacs using musl, it results in a broken Emacs.

This repo is technically a kiss linux overlay

I recommend Kiss linux.

The ever-growing list of Helpful / Credits / Thank You’s

```
1 ;; https://github.com/kisslinux/init/blob/master/lib/init/rc.boot
2 ;; https://github.com/kisslinux/init/blob/master/lib/init/rc.shutdown
3 ;; https://gist.github.com/lunaryorn/91a7734a8c1d93a8d1b0d3f85fe18b1e
4 ;; https://busybox.net/FAQ.html#job_control
5 ;; https://stackoverflow.com/questions/23299314/
   finding-the-exit-code-of-a-shell-command-in-elisp
6 ;; https://github.com/Sweets/hummingbird
7 ;; https://felipec.wordpress.com/2013/11/04/init
8 ;; https://www.emacswiki.org/emacs/PersistentProcesses
9 ;; https://stackoverflow.com/questions/17479529/
   self-contained-portable-emacs
```