

# The Tragedy of systemd

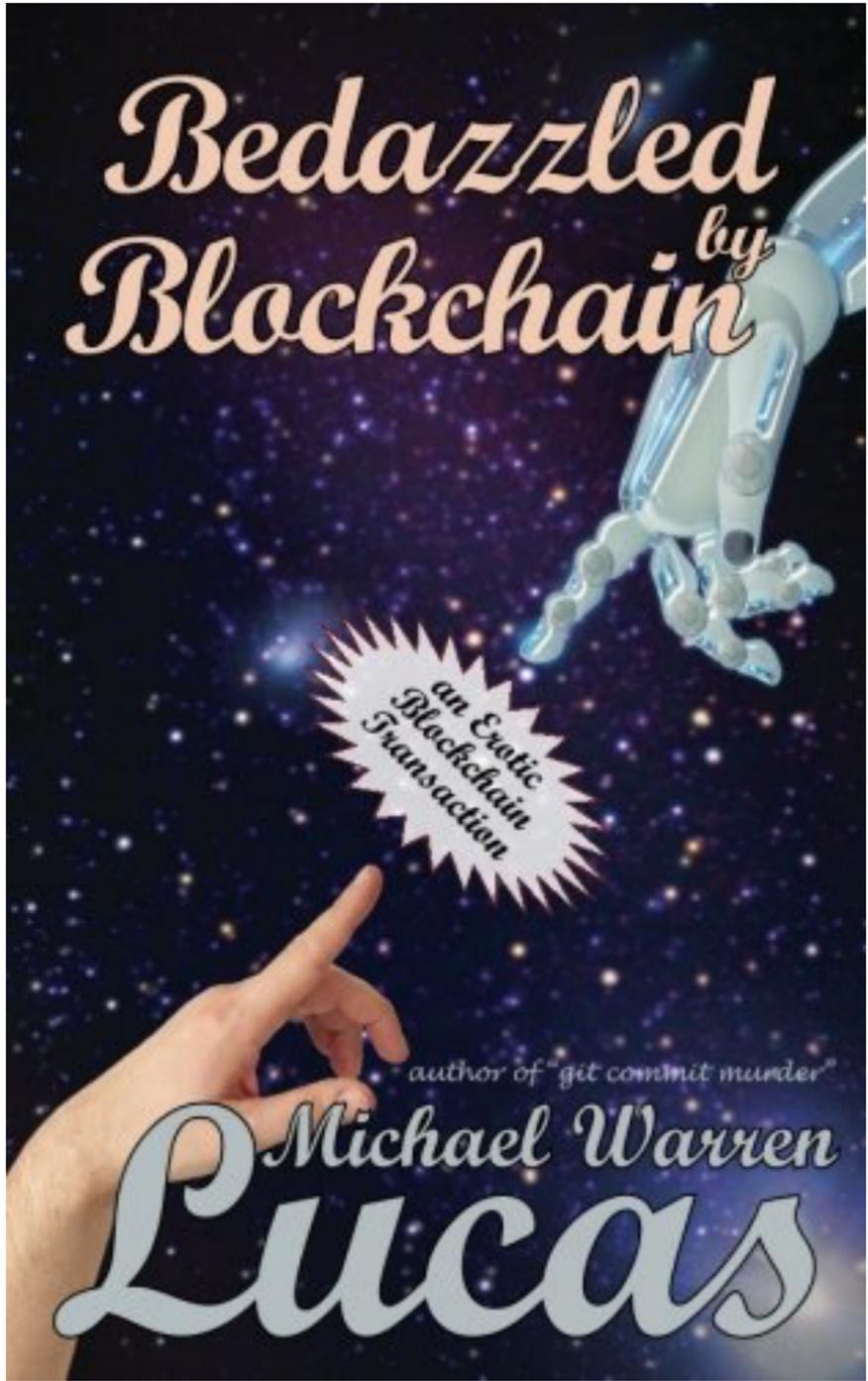
---

*benno@FreeBSD.org*  
*@jeamland*

# The Tragedy of systemd

---

*benno@FreeBSD.org*  
*@jeamland*



# Tragedy

---

From Wikipedia, the free encyclopedia

**Tragedy** (from the **Greek**: τραγωδία, *tragōidia*) is a form of **drama** based on human **suffering** that invokes an accompanying **catharsis** or pleasure in audiences.

**Aurynn Shaw,  
“Contempt Culture”**

*<http://blog.aurynn.com/2015/12/16-contempt-culture>*

**Change**

# The Ancestry of systemd

---

**UNIX**

INIT(8)

INIT(8)

**NAME**

init, rc - process control initialization

**SYNOPSIS**

**/etc/init**  
**/etc/rc**

**DESCRIPTION**

*Init* is invoked as the last step of the boot procedure (see [boot\(8\)](#)). Generally its role is to create a process for each typewriter on which a user may log in.

## **DESCRIPTION**

[...]

When *init* comes up multiuser, it invokes a shell, with input taken from the file */etc/rc*. This command file performs housekeeping like removing temporary files, mounting file systems, and starting daemons.

Then *init* reads the file */etc/ttys* and forks several times to create a process for each typewriter specified in the file. Each of these processes opens the appropriate typewriter for reading and writing. These channels thus receive file descriptors 0, 1 and 2, the standard input, output and error files. Opening the typewriter will usually involve a delay, since the *open* is not completed until someone is dialed up and carrier established on the channel. Then */etc/getty* is called with argument as specified by the last character of the *ttys* file line. *Getty* reads the user's name and invokes *login(1)* to log in the user and execute the shell.

# Seventh Edition Unix

## (1979)

“

... housekeeping functions like...  
mounting filesystems, and starting  
daemons.

- *init(8) manual page, Seventh Edition Unix*

```
$ ps ax
  PID TTY TIME CMD
    0 ? 15749:26 swapper
    1 ? 2:34 /etc/init
24075 co 0:01 -sh
22531 ? 0:00 /etc/init
   15 ? 1:59 /etc/update
   18 ? 6:49 /etc/cron
   744 ? 0:00 /etc/init
   22 ? 0:00 /etc/init
24087 ? 0:00 /etc/init
   24 ? 0:00 /etc/init
24082 05 0:00 - 2
24089 06 0:03 -sh
24066 07 0:01 -sh
24096 06 0:03 ps ax
$
```

*PDP-11/70, Seventh Edition Unix*

```
$ ps ax
  PID TT  STAT   TIME COMMAND
    0  ?  D     5:41 swapper
    1  ?  I     0:30 init
    2  ?  D     0:08 pagedaemon
   35  ?  S    19:48 syslogd
   46  ?  S    58:03 update
   49  ?  I    69:49 cron
   65  ?  S    18:56 routed -q
   70  ?  S    10:37 inetd
   87 co  I     0:00 - Console-1200 console (getty)
 2643 p0  I     0:13 telnetd
 2644 p0  I     0:11 -csh (csh)
13990 p1  S     0:01 telnetd
13991 p1  S     0:04 -sh (sh)
13992 p1  R     0:00 ps ax
$
```

*VAX-11/730, 4.3BSD*

# Living Computers Museum+Labs

*<https://livingcomputers.org>*

**Then things changed**

**Service**

“

... housekeeping functions like...  
mounting filesystems, and starting  
daemons.

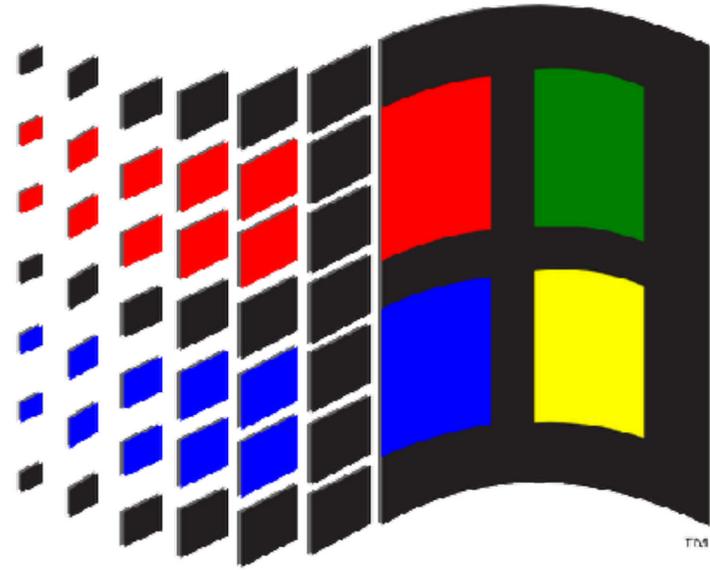
- *init(8) manual page, Seventh Edition Unix*

# System Configuration

# **System Configuration**

# **Service Bootstrap**

# **Automated Service Management**



MICROSOFT  
WINDOWSNT



# The Idea of *systemd*

---

**launchd**

# The Idea of launchd

**From launchd to systemd**

# Lennart Poettering, “Rethinking PID 1”

*<http://Opointer.net/blog/projects/systemd.html>*

“

For a fast and efficient boot-up two things are crucial:

- To start **less**.
- And to start **more** in *parallel*.

*-Lennart Poettering, “Rethinking PID 1”*

“

An init system that is responsible for maintaining services needs to listen to hardware and software changes.

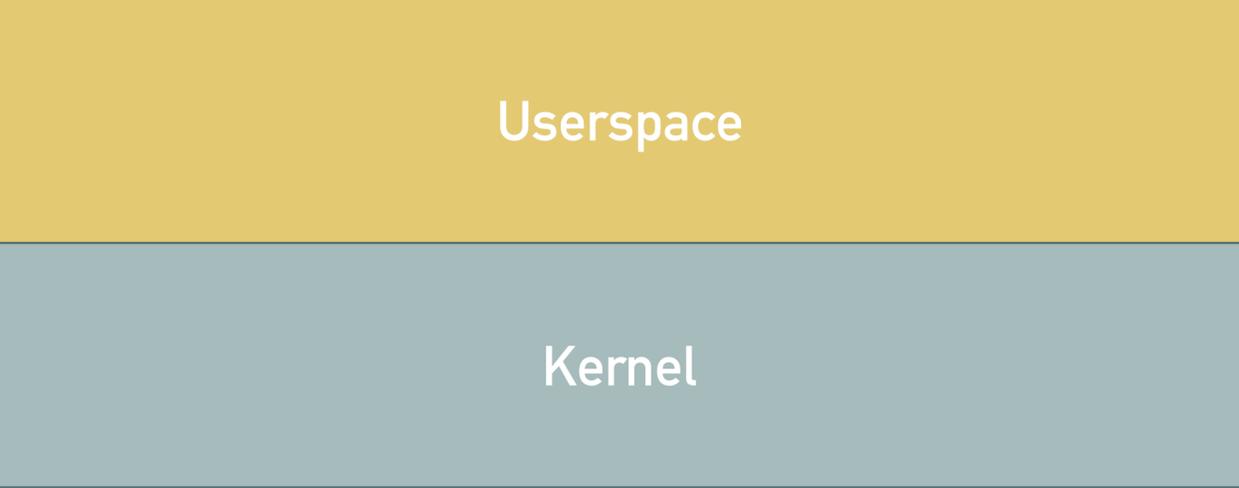
*-Lennart Poettering, “Rethinking PID 1”*

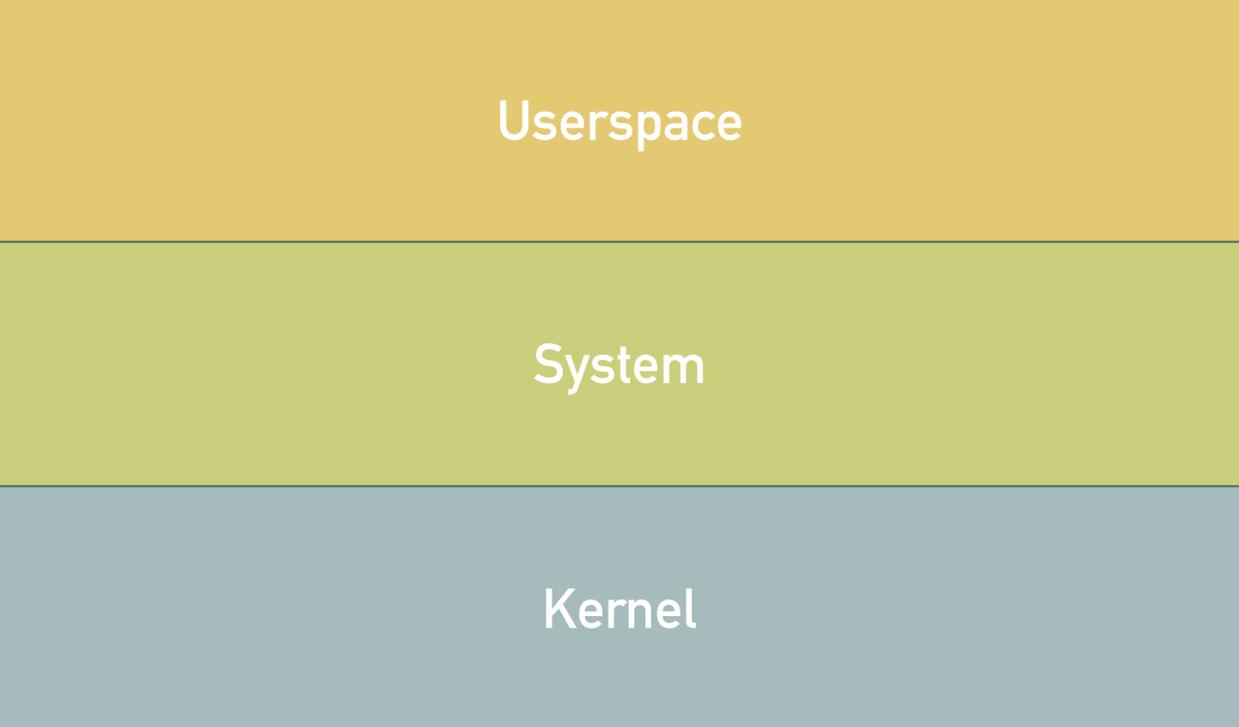
“

[I]s this kind of logic new? No, it certainly is not. The most prominent system that works like this is Apple's launchd system...

*-Lennart Poettering, “Rethinking PID 1”*

# System Management





# The Reality of systemd

---

# Adoption

Fedora 15	May, 2011
openSUSE 12.2	September, 2012
CentOS 7.14.04	April, 2014
Red Hat Enterprise Linux 7.0	June, 2014
SUSE Linux Enterprise Server 12	October, 2014
Debian 8	April, 2015
Ubuntu 15.05	April, 2015

**“It violates UNIX philosophy!”**

**“It’s bloated and monolithic!”**

**“It’s buggy!”**

**“I can’t stand Lennart Poettering!”**

**“It’s not portable!”**

**UNIX is dead**

**cgroups**

**User-level units**

**Change**

# The Tragedy of *systemd*

---

**Change**

**systemd represents change**

# The Knee-jerk

**Abuse Isn't Cool**

**Contempt Isn't Cool**



[#systemd](#) got you down? Come see my talk "Switching to the BSDs" at [@lfnw](#) this weekend.

[linuxfestnorthwest.org/conferences/lf...](https://linuxfestnorthwest.org/conferences/lf...)

1:28 PM - 25 Apr 2018

**Why?**

# The Next Generation

# The Promise of systemd

---

**Picture if you will...**

# Message Transport

# RPC Framework

# Service Lifecycle

# Automation via API

# Containers

# The System Layer

# The Greater Heresies

# Consistent Device Naming

# Better Log/Event/Audit Handling

# A New Model Of An Application

# Catharsis

---

From Wikipedia, the free encyclopedia

**Catharsis** (from Greek κάθαρσις *katharsis* meaning "purification" or "cleansing") is the purification and purgation of emotions—particularly pity and fear—through art or any extreme change in emotion that results in renewal and restoration.