Keeping the kernel relevant with BPF

David S. Miller (Red Hat Inc.)

Kernel APIs Are Hard

You must define the problem being solved

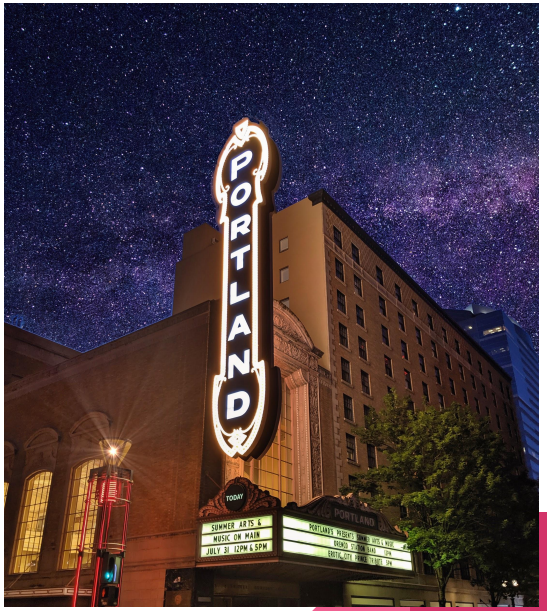
Also, consider existing interfaces

Look into extensibility and future needs

Are there holes in the design

Can it be exploited

Will it be obsolete soon



Kernel Changes Take Time

Then you have to implement it

You have to write tests for it (!!!)

You must propose it upstream

You must address feedback

And repeat all above steps each time

Enterprise distros get it a year later

Kernel APIs Are Confinement

You define the interface and boundaries

And then that’s it

No flexibility

This is inherent in syscall interface structure

The limits are always small and precise

Systems Developers Are Arrogant

To design syscalls properly you must be arrogant

You are putting the caller into a box

You “know better” than the user

Users don’t want to be in a box

Users want to solve their problems

And this may change tomorrow

Users Want Arbitrary Policy

… and they want it now

Maximum flexibility is mandatory

This mean rapid prototyping…

...but for kernel things

By the user

For the user

Kernel Development Pace

New features should be well thought out

Implementation and design should be discussed

Bug fixes should attack the problem not the symptoms

Testing should be complete and rigorous

All of these things take time...

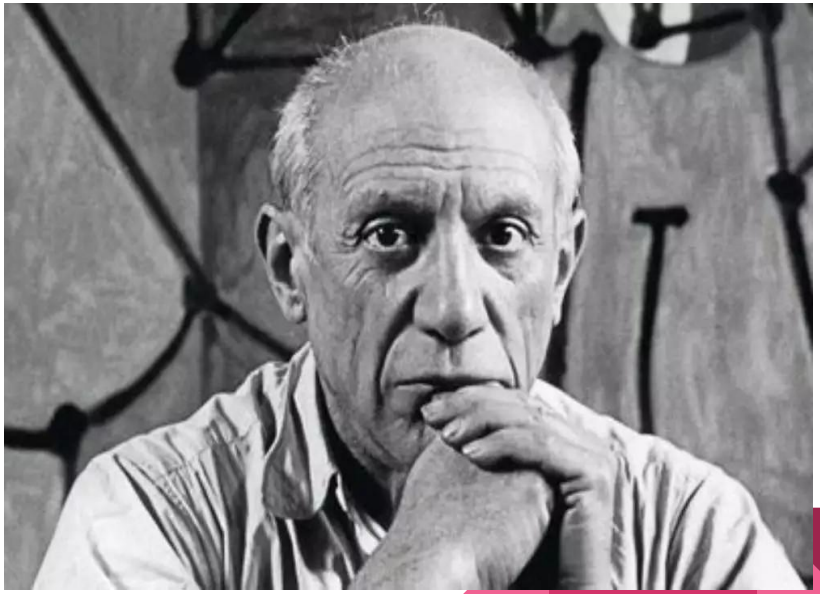
Picasso

Born October 25, 1881

Blue period: 1901-1904

Red Period: 1904-1906

Leading to...



Les Demoiselles d'Avignon

Painted by Picasso around 1907

Widely regarded as the first cubist work

Hidden in Picasso’s studio for some time

Like BPF, people just weren’t ready for it yet

Why is BPF Surprising?

It fundamentally seems contradictory

It provides freedom of expression…

...yet containment and safety

All at the same time

This can’t possibly work

Education is Essential

We need knowledge transfer

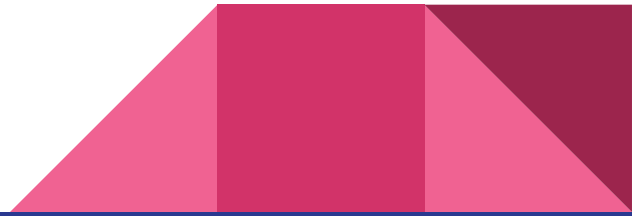
If you understand BPF…

...explain it to other people

If you know someone who understands BPF…

...ask them to teach it to you

Do it for the sake of the kernel



Thank You

Linus Torvalds

Alexei Starovoitov and Daniel Borkmann

Jesper Dangaard Brouer

And all the sad cats...