



Bridging the Linux Tux and the FreeBSD Beastie : Linuxulator, LinuxKPI, and LLM-Assisted Porting Challenges on License Issue



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA / OCT. 5-7, 2026

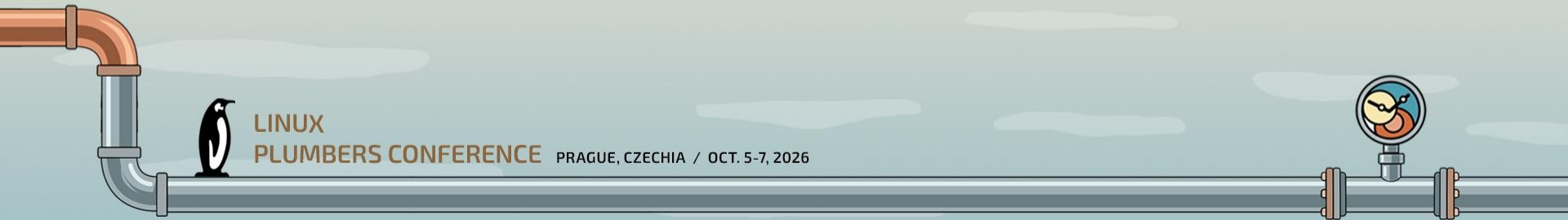
People

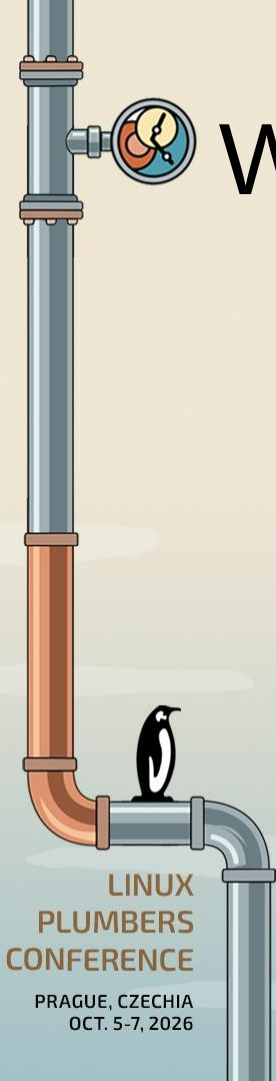
Ed Maste, senior director
Li-Wen Hsu, software engineer
FreeBSD Foundation

“Ruinland” ChuanTzu Tsai,
advanced engineer
Andes Technology

Details :

<https://freebsd.foundation.org/about-us/our-team/>



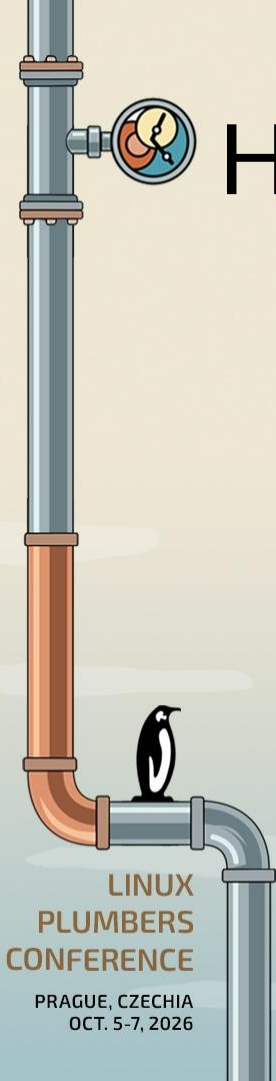


What are these ?

Linuxulator : Linux usermode binary compatibility layer - - to run unmodified Linux binaries on top of FreeBSD
<https://wiki.freebsd.org/Linuxulator>

LinuxKPI : the headers and function implementation in FreeBSD kernel to “kinda” make Linux kernel drivers to be easier to ported to FreeBSD
<https://wiki.freebsd.org/LinuxKPI>

Great slides from people working on BATMAN for FreeBSD :
<https://indico.bsdcan.org/event/1/contributions/12/attachments/14/15/presentation.pdf>



How it (Linuxualtor) works, exactly

In FreeBSD kernel, there's a mechanism call image activation (imgact), basically the equivalence of Linux's "binfmt."

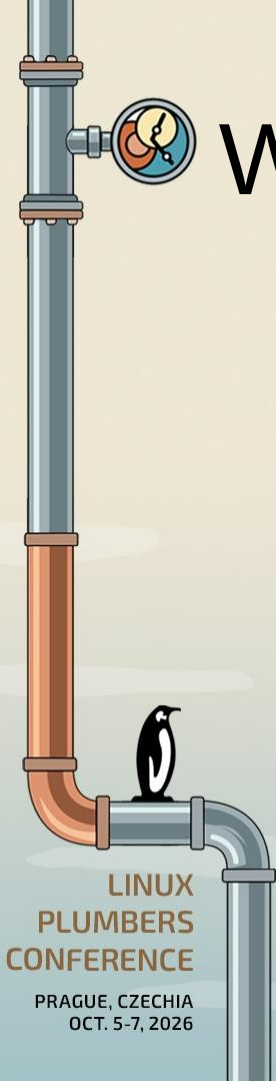
And for imgact of ELF, there's a concept in FreeBSD - - "brand."

For different "brands" of ELF, FreeBSD kernel allows people to hook certain functions to provide tweaks for compatibility - - for instance, the syscall ABI.

For FreeBSD on RISC-V, syscall num is \$a0. But on Linux, it's \$a7. And don't forget that Linux has architecture syscalls, which RISC-V Linux have 2 of them : hwprobe and icache_flush.

LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026



What is done recently on RISC-V ?

This work is done by Li-Wen Hsu from FreeBSD Foundation.

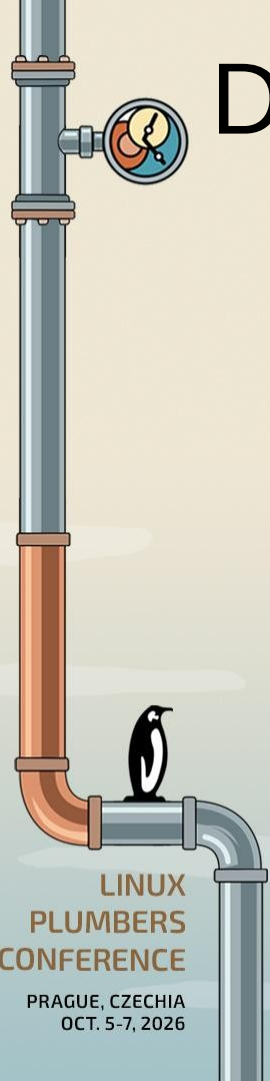
The PoC you are seeing is running on Andes' Qilai CPU.

LINUXTEST from FreeBSD has passed 103 tests out of 105.
The 2 failed ones are related to vDSO.

Basic Debian rootfs shell is working.
Even a simple Golang hello world could be compiled and it runs !

LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026



Details

The key transition is setting `p->p_sysent = &elf_linux_sysvec`: this tells the existing FreeBSD execution and trap machinery to use Linux ABI handlers for that process.

Linux64 module initialization registers Linux ELF support.

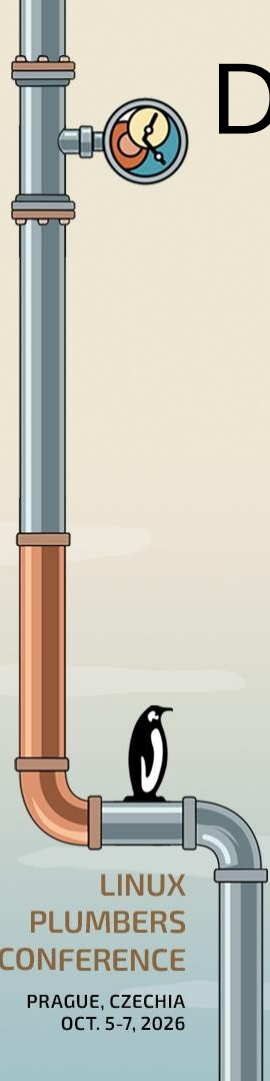
In `sys/riscv/linux/linux_sysvec.c`:

`linux64_elf_modevent(MOD_LOAD)` calls `elf64_insert_brand_entry()` for the glibc brands (it's libc dependent).

`EM_RISCV` and `elf_linux_sysvec`.

`linux_vdso_install()` prepares the vDSO

`linux_exec_sysvec_init()` calls `exec_sysvec_init()` and connects the vDSO to shared timekeeping data.



Details

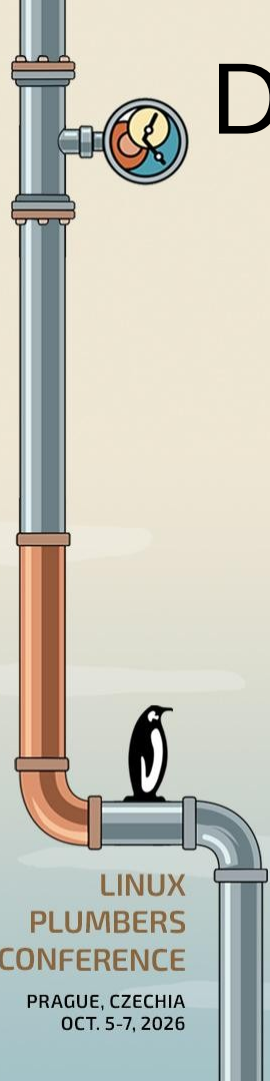
When shell requests execution through native `execve()`

In `sys/kern/kern_exec.c`

`sys_execve()`

- `pre_execve()`
- `exec_copyin_args()`
- `kern_execve()`
- `do_execve()`

The kernel copies the pathname, arguments and environment, resolves the executable, checks access, and examines its image header.



Details

The kernel creates the new address space and Linux process state.

Inside `exec_elf64_imgact()`:

`exec_new_vmpace(imgp, sv)`

→ `sv->sv_onexec`

→ `linux_on_exec_vmpace()`

→ `linux_map_vdso()`

→ `linux_on_exec()`

→ `linux_proc_init()`

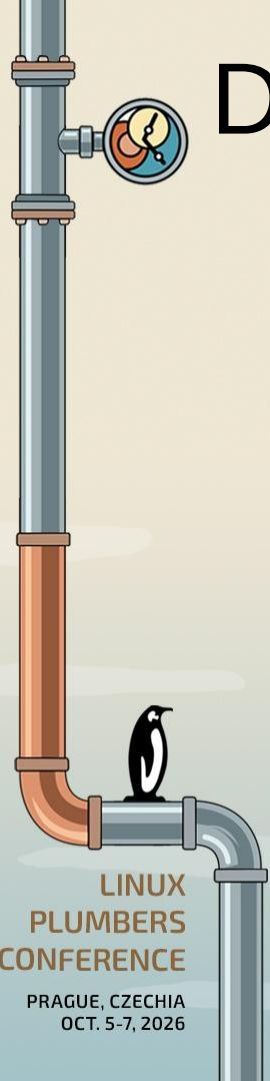
→ `linux_pwd_onexec()` [**switching from native to Linux**]

This replaces the old mappings, maps the Linux vDSO, initializes Linux emulation bookkeeping, and establishes the ABI root directory, normally `/compat/linux`.

After `exec_new_vmpace()` returns, the ELF activator assigns:

```
imgp->proc->p_sysent = sv;    /* &elf_linux_sysvec */
```

```
imgp->proc->p_elf_brandinfo = brand_info;
```



Details

The process's Linux syscall vector selects the handler.

In `syscallenter()`:

```
p->p_sysent->sv_fetch_syscall_args
```

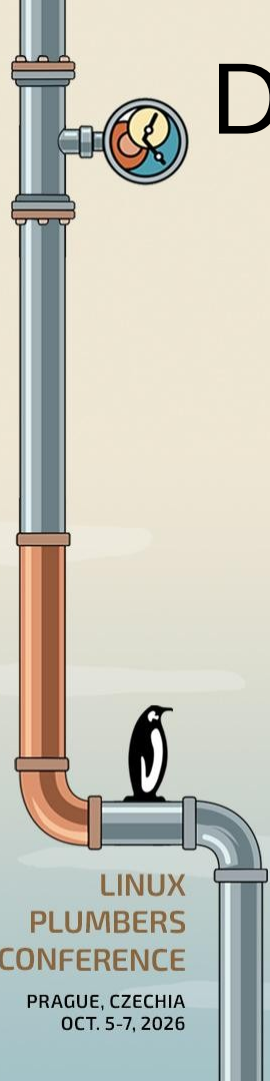
```
→ linux_fetch_syscall_args()
```

```
→ linux_sysent[a7]
```

`linux_fetch_syscall_args()` reads `a7`, copies `a0–a5`, and selects the table entry. An out-of-range number selects `nosys_sysent`.

`syscallenter()` then invokes the selected function through:

```
error = (se->sy_call)(td, sa->args);
```



Details

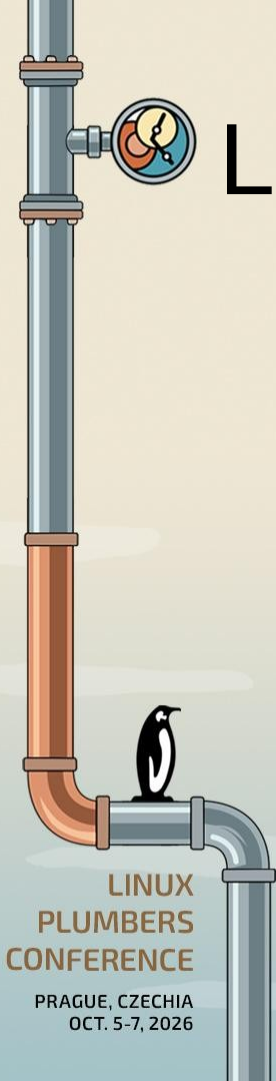
The handler also translates the request into FreeBSD operations.

For example, Linux write, syscall number 64, follows:

```
linux_sysent[64]  
→ linux_write()  
→ sys_write()  
→ kern_writev()  
→ dofilewrite()
```

linux_write() adapts the arguments and applies Linux-specific error handling. The actual I/O uses FreeBSD's file machinery.

Other calls may use shared Linuxulator handlers or compatible FreeBSD.



LinuxKPI in general

Targeting roughly at Linux 5.15 so as to port DRM and 802.11 drivers.

Due to the nature of Linux kernel interface, this is rather flaky.



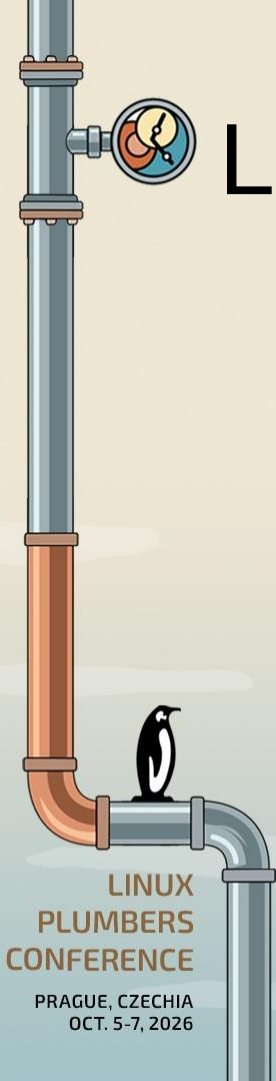
LinuxKPI for RISC-V

Not yet :-p



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026



LLM assisted work

A Marvel NIC driver has been ported to FreeBSD and merged :

<https://wiki.freebsd.org/DevSummit/202603?action=AttachFile&do=view&target=aq2.pdf>

Currently, Marvel is not complaining. But no one can guarantee it will be okay with other drivers and different parts of Linux kernel.

License issue is a huuuuuuuuuge problem.

LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026



Not a free-rider.

FreeBSD Foundation has contributed quite some works to Linux community :

<https://gist.github.com/lwhsu/a6fed0e90a645212ad2261b1f05dfe74>

Commits with author email @freebsd.org: 50 (12 distinct people)

Commits by the same people under personal email addresses: 12 kept

Hotspot : iwlwifi



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026



What is needed

Linux is the de-facto dominator and standard in the non-Windows world. People tend to align with Linux for real work. Even Windows has WLS now.

License agreements or co-ordination would be nice.



LINUX
PLUMBERS
CONFERENCE

PRAGUE, CZECHIA
OCT. 5-7, 2026



Backup

Lorem ipsum.



**LINUX
PLUMBERS
CONFERENCE**

PRAGUE, CZECHIA
OCT. 5-7, 2026