# my mainframe runs arch btw

porting Arch Linux to IBM s390x

Josephine Pfeiffer

FOSDEM 2026

## the premise

I use Arch

## the premise

I use Arch (btw)

## the premise

I use Arch (btw)

on an IBM mainframe

## the premise

I use Arch (btw)

on an IBM mainframe

an architecture lineage from 1964

## s390x in 30 seconds

- IBM Z / IBM LinuxONE
- 64-bit big-endian
- boots via **IPL**
- your bank runs on this

## “why?”

## “why?”

I thought it would be funny :p

## the problem

Arch Linux assumes you build entirely on the target architecture

## the problem

Arch Linux assumes you build entirely on the target architecture

I do not own a mainframe

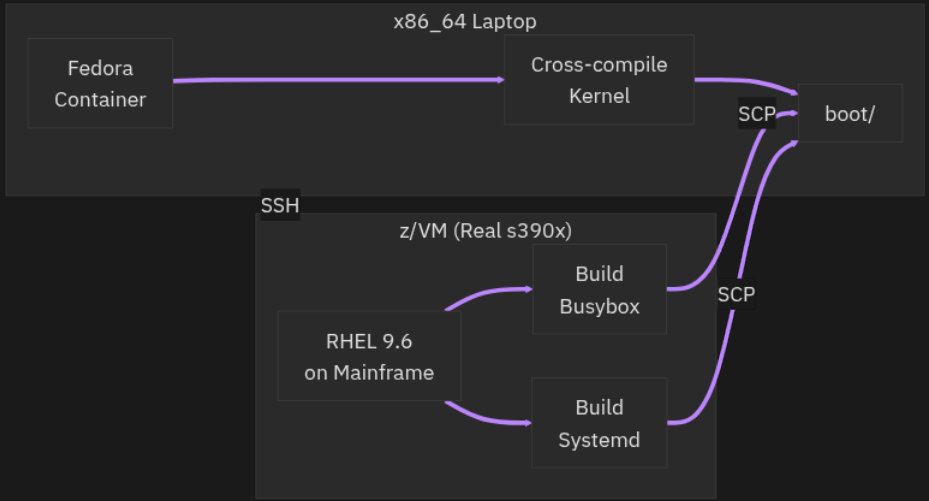
## the problem

Arch Linux assumes you build entirely on the target architecture

I do not own a mainframe

(but I have access to one)

# the solution: a hybrid build



## step 1: the kernel

cross-compile with `s390x-linux-gnu-gcc` in a Fedora container

```
make ARCH=s390 CROSS_COMPILE=s390x-linux-gnu- \
     -j$(nproc) bzImage modules
```

this part actually works fine

## step 1: the kernel

cross-compile with `s390x-linux-gnu-gcc` in a Fedora container

```
make ARCH=s390 CROSS_COMPILE=s390x-linux-gnu- \
     -j$(nproc) bzImage modules
```

this part actually works fine

this is the last time I will say that

## step 2: busybox

busybox must be statically linked for the initramfs

## step 2: busybox

busybox must be statically linked for the initramfs

reliably static-linking s390x binaries requires native compilation

## step 2: busybox

busybox must be statically linked for the initramfs

reliably static-linking s390x binaries requires native compilation

you need a real mainframe

```
$ ssh mainframe
$ make CONFIG_STATIC=y
$ scp busybox-s390x-static laptop:boot/
```

2.3 MB. 60+ applets

## step 3: mkinitcpio

Arch’s initramfs generator assumes it can *run* the binaries it packs

## step 3: mkinitcpio

Arch’s initramfs generator assumes it can *run* the binaries it packs

it cannot run s390x binaries on x86_64.

## step 3: mkinitcpio

Arch's initramfs generator assumes it can *run* the binaries it packs

it cannot run s390x binaries on x86_64.

```
# Can't do this on x86_64:
busybox --list
# So instead: hardcode all 60+ applet names
local busybox_applets=(
    "[" "[[" "ash" "awk" "basename" "cat"
    "chmod" "cp" "dd" "df" "echo" "env"
    # ... 50 more lines of this
)
```

## step 4: the CONFIG_UNIX incident

systemd boots.

## step 4: the CONFIG_UNIX incident

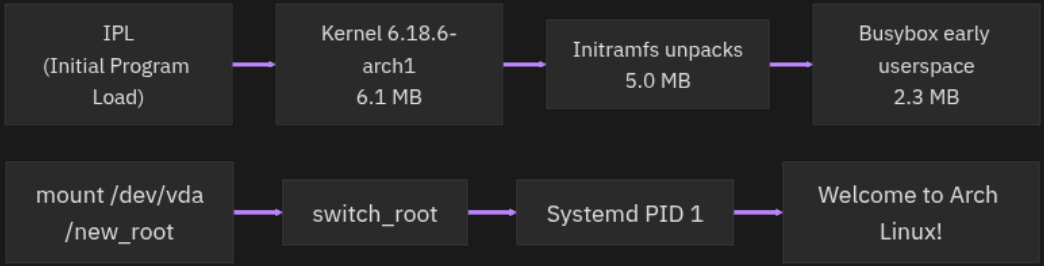
systemd boots. systemd immediately dies.

## step 4: the CONFIG_UNIX incident

systemd boots. systemd immediately dies.

```
CONFIG_UNIX=y
# AF_UNIX sockets. systemd needs these.
```

## the boot sequence




total system image: **11 MB** smaller than most Electron apps.

## it works :D

```
$ uname -a
Linux archlinux 6.18.6-arch1-1-s390x #1 SMP s390x GNU/Linux

$ cat /etc/os-release
NAME="Arch Linux"
PRETTY_NAME="Arch Linux"

$ busybox | head -1
BusyBox v1.37.0 (s390x) multi-call binary.

$ systemctl status
  State: running
  Units: 12 loaded
```

## what I learned

- cross-compilation has hard limits so sometimes you need the real hardware
- distribution tooling has architecture assumptions everywhere
- `CONFIG_UNIX=y`

## Arch Linux on s390x

**working today:**

- kernel, initramfs, busybox, systemd
- full boot to systemd shell
- QEMU emulation + real z/VM hardware

**next:**

- pacman and makepkg for s390x
- pacstrap and base packages

## thank you

github.com/pfeifferj/archlinux-s390x

josie.lol