Streamlining kernel hacking with mkosi-kernel
About Me

- Daan De Meyer
- systemd/mkosi maintainer
- Linux Userspace Team @ Meta
Motivation

- How do I test my patch without bricking my machine?
- How do I quickly set myself up for kernel hacking on a new machine?
- How do I get a fast edit => compile => test cycle when hacking on the kernel?
- How do I integrate various kernel related userspace projects?
What is mkosi?
Quick Start

mkosi -d arch -p systemd -p linux --autologin qemu
CentOS Stream 9
Kernel 5.14.0-350.el9.x86_64 on an x86_64
debug login: root (automatic login)
Last login: Wed Aug 9 12:44:36 on tty1
[root@localhost ~]#  

localhost login: root (automatic login)

Have a lot of fun...

localhost:~ #  

Arch Linux 6.4.8-arch1-1 (ttyS0)
archlinux login: root (automatic login)
[root@archlinux ~]#  

Ubuntu 23.04 localhost ttyS0
localhost login: root (automatic login)
Welcome to Ubuntu 23.04 (GNU/Linux 6.2.0-1009-kvm x86_64)
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage
Last login: Wed Aug 9 10:25:24 UTC 2023 on ttyS0
root@localhost:~ #  

Debian GNU/Linux trixie/sid localhost ttyS0
localhost login: root (automatic login)
Linux localhost 6.4.0-1-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.4.4-2 (2023-07-30) x86_64
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Aug 9 10:30:29 UTC 2023 on tty1
root@localhost:~ #  

Fedora Linux 39 (Rawhide Prerelease)
Kernel 6.5.0-0.rc5.36.fc39.x86_64 on an x86_64 (ttyS0)
fedora login: root (automatic login)
[root@fedora ~]#  
Configuration

[Match]
Distribution=fedora

[Distribution]
Release=rawhide

[Content]
Packages=kernel-core
        systemd
        systemd-boot
        udev
        util-linux
        grub2-pc
Using mkosi for kernel development
mkosi-kernel

- A mkosi configuration for hacking on the kernel and related userspace projects
Getting started

```bash
git clone https://github.com/systemd/mkosi
ln -s $PWD/mkosi/bin/mkosi /usr/local/bin/mkosi

git clone https://github.com/DaanDeMeyer/mkosi-kernel

cd mkosi-kernel

tee mkosi.local.conf <<EOF

[Distribution]
Distribution=fedora

[Config]
@include=modules/kernel

[Content]
BuildSources=..:/kernel:kernel

EOF

mkosi -f qemu
```
mkosi.kernel.config

- A minimal kconfig for fast kernel builds that will boot with “mkosi qemu”
  - Enable features
  - Keep most drivers disabled

Build and install (some) selftests

```
Environment=SELFTESTS=1
  SELFTESTS_TARGETS= ...
  SELFTESTS_SKIP_TARGETS= ...
```

Use a custom kbuild config

```
Environment=CONFIG=<path-to-config>
```

Boot without initramfs

```
CONFIG_VIRTIOFS=y
```

Make

```
make KCONFIG_ALLCONFIG=$CONFIG alldefconfig
```
Useful settings

Mount extra directories into the VM
RuntimeTrees=../kernel:kernel

Add ephemeral 10G nvme disk to VM
QemuDrives=btrfs:10G::aio=io_uring
QemuArgs=-device nvme,serial=btrfs,drive=btrfs

Add extra files to the image
ExtraTrees= ...

Extra kernel command line arguments
KernelCommandLineExtra=panic=1
 ... oops=panic

Pass in your own kernel
QemuKernel=<path-to-kernel>

Boot with UEFI firmware
QemuFirmware=uefi
SSH over VSOCK

```
Host unix/* vsock/*
    ProxyCommand {{LIBEXECDIR}}/systemd-ssh-proxy %h %p
    ProxyUseFdpass yes
    CheckHostIP no

    # Disable all kinds of host identity checks, since these addresses are generally ephemeral.
    StrictHostKeyChecking no
    UserKnownHostsFile /dev/null
```
Running integration tests with mkosi
What about virtme-ng?

- Similar goals, different approaches

- **mkosi**
  - Always builds a custom image
  - Provides a more full featured VM
  - systemd as init
  - General purpose

- **virtme-ng**
  - Uses host rootfs or prebuilt rootfs
  - Spawns micro VM without ACPI/PCI/...
  - Custom init
  - Focused on kernel development
Reactions from users

First day using mkosi-kernel for testing 11:19 AM
Exciting! Let's hope it works 11:20 AM
it's very nice 11:24 AM

Josef Bacik

alright, lets boot this new mount api kernel

loving mkosi for this shit

Josef Bacik

alright fine i'll finally sit down and move all my shit over to mkosi
this thing is amazing
https://github.com/systemd/mkosi
https://github.com/DaanDeMeyer/mkosi-kernel