Hermetic Builds with Bazel

Matthias Männich <maennich@android.com>
TL;DR: What?

$ BUILD_CONFIG=common/build.config.gki.aarch64 build/build.sh

$ bazel build //common:kernel_aarch64

https://source.android.com/docs/setup/build/building-kernels#building-with-bazel
Android Kernel Builds

"some document" -> "build.sh" -> Bazel (Kleaf)
Android Kernel Builds

- build.sh
  - build.config configuration
  - build
  - package
  - module builds
  - mixed builds
  - ...

Android Kernel Builds

- build.sh
  - lots of control via env variables
  - ever-growing shell script collection
  - difficult to maintain
  - hermeticity / reproducibility as hack
  - limited parallelism
Bazel

● Scalable
● Built-in sandboxing for hermetic builds
● Fast incremental builds
● Build dependency analysis
● Parallelism
● Remote Build Execution (RBE)
● Future Android Platform build system

https://bazel.build
Kernel Builds with Bazel (Kleaf)

- Consistent environment setup
- Hermetic toolchain (tools, compilers, etc.) enforced
- Wraps make defconfig and make steps
- Kbuild remains authoritative build system
Kernel Builds with Bazel

```python
load("//build/kernel/kleaf:kernel.bzl", 
    "kernel_build", "kernel_module")

kernel_build(
    name = "kernel",
    outs = ["vmlinux"],
    build_config = "common/build.config.gki.aarch64",
    srcs = glob(["**"]),
)

kernel_module(
    name = "nfc",
    srcs = glob(["**"]),
    outs = ["nfc.ko"],
    kernel_build = "/common:kernel",
)
```

**BUILD.bazel**

- uses build.config (during migration)
- defines GKI kernel build
- kernel module against GKI
- consistent hermetic toolchain (enforced)
- sandboxed build

https://android.googlesource.com/kernel/build/+/refs/heads/master/kleaf/docs/kleaf.md
Kernel Builds with Bazel

- Builds "just enough" kernel
- Ensures limited visibility
  - kernel headers
  - scripts/
  - no other source files (.c)
- Enforces compatible toolchain
- Requires Kbuild makefiles (Kbuild, Makefile, Kconfig)

kernel_build(name = "kernel", ...)
kernl_module(name = "nfc", ...)

$ bazel build nfc

https://android.googlesource.com/kernel/build/+refs/heads/master/kleaf/docs/kleaf.md
Driver Development Kit (DDK)

- **Generates** Kbuild makefiles (Kbuild, Makefile, Kconfig)
- Ensures correct build dependencies
- Restricts source file visibility to declared inputs

```python
ddk_module(
    name = "base",
    srcs = [
        "base.c",
        "impl_base.h",  // available for module local compilation
    ],
    hdrs = ["base.h"],  // exported to other modules
    kernel_build = "/common:kernel_aarch64",
)

ddk_module(
    name = "nfc",
    srcs = [
        "nfc.c",          // can #include base.h and
        "nfc.h",          // use symbols exported from base
    ],
    kernel_build = "/common:kernel_aarch64",
    deps = ["base"],
)
```

DRAFT API

https://android.googlesource.com/kernel/build/+refs/heads/master/kleaf/docs/kleaf.md
Useful Build Flags

- `--lto={default,none,thin,full}`: LTO mode
- `--use_prebuilt_gki=8728676`: Download prebuilt GKI from ci.android.com
- `--config=local`: Local make cache (breaks sandbox)
- `--config=fast`: --config=local and --lto=thin
- `--config=release`: SCM versions, used on ci.android.com
- `--kasang`: Enable KAsan
- `--debug_print_scripts`: Print assembled scripts that get executed

https://android.googlesource.com/kernel/build/+/refs/heads/master/kleaf/docs/kleaf.md
Migrating to Kleaf

- `build.sh` deprecated
- automatic command line migration
- automatic `build.config` migration
Questions?