linaro Connect 2025

Page-agnostic AOSP on devboards - Enablement experience

Amit Pundir

Agenda

- What is page-agnostic AOSP?
- Android support for page-agnostic builds
- Enablement on devboards
- Challenges
- App Observations
- CI enablement in Linaro Android Testing

What is Page-Agnostic AOSP

- Page-agnostic means applications don’t depend on the page size
  - OS doesn’t make page size assumptions
  - Same application can run on 4kb / 16kb page-size devices
- Benefits
  - When used with 16kb page_size kernel: **5-10%** overall performance boost at ~9% additional memory cost

- Details at the AOSP Blog about 16k page-size support in Android

Android support for page-agnostic builds

linaro Connect

- OS Changes
  - Compile time PAGE_SIZE macro replaced with runtime enabled getpagesize()
  - OS Binaries are 16kb page aligned
  - OS binaries are built with separate loadable segments
- Filesystem changes
  - Most performant combination will have same FS block size as page size
  - For compatibility, some file systems support different page size enablement
    - **ext4, erofs** when built with 4k block size, will boot with both 4k and 16k page size kernels
    - **F2fs** only supports block size = page size
    - => For page_agnostic builds supporting both 4k and 16k, only ext 4 and erofs can be used

Enablement on Devboards

- Build support for page agnostic builds
  - PRODUCT_NO_BIONIC_PAGE_SIZE_MACRO := true
  - PRODUCT_MAX_PAGE_SIZE_SUPPORTED := 16384
- **64bit_only** is a prerequisite
- Kernel build
  - With KLEAF: --page_size=16k
  - Direct kernel build: set CONFIG_ARM64_16K_PAGES instead of CONFIG_ARM64_4K_PAGES
- Mesa updates for 16k
  - Needed to be rebuilt with -Wl,-z,max-page-size=16384 flags
- FS support update
  - In accordance with the compatibility guideline, we use only ext4 and erofs for page agnostic builds

Challenges

- GPU crash
  - Root-caused to an Adreno smmu issue on RB3
- Webview
  - During our development, AOSP still had old webview component, built for 4k support
  - Resulted in crash of chrome on 16k
  - Reported and fixed in AOSP
- ABL page size issue
  - ABL cannot unpack 16k compressed images
  - Needed to pack the built image with 4k page size explicitly
    - BOARD_MKBOOTIMG_ARGS += --pagesize 4096
    - INTERNAL_VENDOR_BOOTIMAGE_ARGS += --pagesize 4096

App Support

- Not all apps work well with 16kb kernels
  - In some cases, the 16kb back compat option can help
  - To enable for all apps:
    - adb shell setprop bionic.linker.16kb.app_compat.enabled true
    - adb shell setprop pm.16kb.app_compat.disabled false
  - For specific app:
    - Toggle the android:pageSizeCompat property in the AndroidManifest.xml as needed

- Tested a few top rated apps to check their 16kb readiness
  - Some top rated apps from F-droid store and a few direct downloads from the developer(s)
  - 3dMark, F-droid, Jellyfin, K-9, Signal, Tor browser, VLC, WhatsApp

App Observations

linaro Connect

- Assessment findings
  - All the apps work on 4kb page_size build
  - F-droid, Jellyfin, K-9, WhatsApp apps work on 16kb build
    - check_elf_alignment.sh: ELF segments of the shared libraries are 16kb aligned
  - VLC works in 16kb page size compatibility mode
  - 3dMark switched to 16kb compat mode but crashes later-on
  - Signal, Tor browser fails to start

} *ELF segments are not 16kb aligned*

    - Current guess about the failures is that the apps make some page size assumptions in the code itself

CI enablement in Linaro Android Testing

- Enabled with android-mainline + AOSP/main combinations
- Devices supported: RB3, RB5, SM8550-HDK
- ~200 reports sent out

linaro Connect 2025

Thank You!

Contact us at:
amit.pundir@linaro.org
sumit.semwal@linaro.org
IRC: pundir, sumits on #linaro-android, #aosp-developers IRC channels at OFTC.net