ANDROID KERNEL CUSTOMIZATION

The operating system of a device is the part of the device responsible for basic use and administration. This includes the kernel and device drivers, boot loader, command shell or other user interface, and basic file and system utilities. Whereas the user interface is the outermost portion of the operating system, kernel is the innermost. It is the core internals, the software that provides basic services for all other parts of the system, manages hardware and distributes system resources.

Typical components of a kernel are interrupt handlers to service interrupt requests, a scheduler to share processor time among multiple processes, a memory management system to manage process address spaces, and system services like networking and interprocess communication. On modern systems with protected memory management units, the kernel typically resides in an elevated system state as compared to normal user applications. This includes a protected memory space and full access to hardware. This system state and memory space is collectively referred to as kernel-space. Conversely, user applications reside in user-space.

Applications running on the system communicate with the kernel via system calls. An application typically calls functions in a library-Eg The C library–that in turn rely on the system call interface to instruct the kernel to carry out tasks on the application's behalf.

BUILDING REQUIREMENTS

- Linux machine
- Kernel Source Code (Device Compatible)
- Vanilla/Custom Toolchain
  - Linaro
  - UBERTC
  - SaberMod

TOOLCHAIN

A toolchain is a set of distinct software development tools that are linked (or chained) together by specific stages such as GCC, binutils and glibc (a portion of the GNU Toolchain). Optionally, a toolchain may contain other tools such as a Debugger or a Compiler for a specific programming language, such as ,C++. Quite often, the toolchain used for embedded development is a cross toolchain, or more commonly known as a cross compiler. All the programs (like GCC) run on a host system of a specific architecture (such as x86) but produce binary code (executables) to run on a different architecture (e.g. ARM). This is called cross compilation and is the typical way of building embedded software. It is possible to compile natively, running gcc on your target. Before searching for a prebuilt toolchain or building your own, it's worth checking to see if one is included with your target hardware's Board Support Package (BSP) if you have one.

CUSTOMIZATION

- Governors

  A governor is actually nothing more than a behavior profile for your CPU, the governor will tell the CPU exactly what to do in what situation. The term ‘governor’ has nothing to do with it’s function, imagine it was called CPU Presidents! sounds strange but it’s as normal as using the word Governor.
  Why is it important to have custom governors: The default governors added to kernels are “ondemand, powersave, performance, conservative” and optionally “userspace”, all these governors are pretty basic and not optimized for usage/battery whatsoever.

- I/O Schedulers

  Next to adding custom CPU governors, you can also enhance your kernel by adding new I/O Schedulers if needed. Globally Governors and Schedulers are the same, they both provide a way how the system should work. but on Schedulers it’s all about the Input/Output datastream except the CPU settings. the I/O Schedulers decide how the upcoming I/O activity is being scheduled.The standard schedulers such as “noop” or “cfq” are pretty decent actually. They perform very reasonable.

- Overclocking/Underclocking

  Overclocking is the configuration of a computer hardware component to operate at a faster rate than was certified by the original manufacturer, generally specified as a given clock frequency in megahertz (MHz) or gigahertz (GHz). Commonly the operating voltage of the overclocked device is also increased, which can help with maintaining the component's operational stability at the accelerated speeds. A given semiconductor device, however, will generate more heat when operated at higher frequencies and voltages, so most overclocking attempts will increase power consumption and heat as well. The overclocked device may be unreliable or fail completely if the additional heat load is not removed, or if the supporting power delivery components cannot handle the increased power demands.

OTHER CUSTOMIZATIONS

- Camera Sensor
- Sound Control
- exFAT and NTFS support
- Dynamic Fsync
- ZSWAP optimization
- Touch boost
- LZ4 compression/decompression drivers
- Full control over thermal parameters
- KCAL control
- Vibration control

- ZRAM, ZSMALLOC optimizations
- TCP congestion algorithm optimization

REFERENCES

[1] IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 2, No 3, March 2013

[2] Wainner, Scott; Robert Richmond (2003). The Book of Overclocking. No Starch Press. Pp. 1–2. ISBN 1-886411-76-X.

[3] "Toolchains". elinux.org. 2013-09-08. Retrieved 2013-10-21.

[4] “Adding features to your kernel”. Xda-university.com © 2016 XDA-University

[5] Bidinger, Matt (2011-09-13). "AMD Bulldozer Breaks CPU Frequency World Record". Retrieved 2016-01-08.