
1 aa-kernel

`aa-kernel` is an embedded operating system for 32 and 64-bit ARM CPUs. The kernel consists of:

- A base set of subsystems and supporting code, located in `src/boot` and `src/kernel`.
- Architecture and platform-specific subsystems, located in `src/arch` and `src/platform`.
- Drivers not limited to any one architecture or platform, such as the SDHCI driver, located in `src/drivers`.

1.0.1 Design Goals

Much of the work on `aa-kernel` was carried out in order to support FPGA development on Zynq systems. As a result, the Cortex A9 port is currently the most well-supported variant in terms of driver support and testing.

Physical testing for this port was carried out on the Digilent Zybo board. The kernel is also capable of booting on the aarch64 virt platform, the Integrator CP, and the Raspberry Pi, though the Zynq and virt ports are the only currently tested ports.

1.0.2 Boot Sequence

On boot, on-board firmware or an off-board JTAG command sequence loads the kernel image into memory and, on a single core, jumps to the `_start` symbol, defined in `src/arch/{arch}/start.s`. This `_start` symbol is an assembly function that sets up the stack pointer, installs the base interrupt vectors, and calls the `init()` function located in `src/boot/init.c`. `init()` executes architecture-specific initialization functions (defined in the relevant `arch` folders) and initializes the buddy-block page allocator, slab allocator, random allocator, thread subsystem and IRQ subsystem. It then runs `platform_init()`, also defined in a specific `platform` folder, enables IRQs, and finally runs `kmain()`.

1.0.3 Subsystems

IRQs (src/kernel/irq.c)

IRQs refer to any interrupt generated externally, such as from hardware. The base IRQ handler structure is typically installed at the start of physical memory for ARM CPUs.

Timers

Interrupt-driven timers enable scheduling events in time, allowing features such as preemptive multitasking. A preset timer IRQ is defined per-platform in

src/include/platform/{platform}/irq_num.h

and installed during platform_init() at boot.

Buddy-Block Allocator (src/kernel/bb_alloc.c)

The buddy-block allocator is the base of all other allocators and allows allocating 2^n pages of physically contiguous, page-aligned blocks of memory.

Slab Allocator (src/kernel/bb_alloc.c)

In many portions of the kernel, data structures of a certain fixed size are repeatedly allocated during runtime. The slab allocator dedicates a block of memory to contiguous arrays of these

Random Allocator (src/kernel/malloc.c)

The random allocator defines void *malloc(uint32_t size), which allocates size bytes successfully or returns 0.

Block Devices (src/kernel/block_cache.c)

dank

Console (src/kernel/console.c)

shit

Multitasking (src/kernel/thread.c)

1.0.4 Drivers

PL011 Serial Console (src/drivers/serial/pl011.c)

PL390 Interrupt Controller (src/drivers/serial/pl011.c)

SD Host Controller (src/drivers/block/sdhci.c)