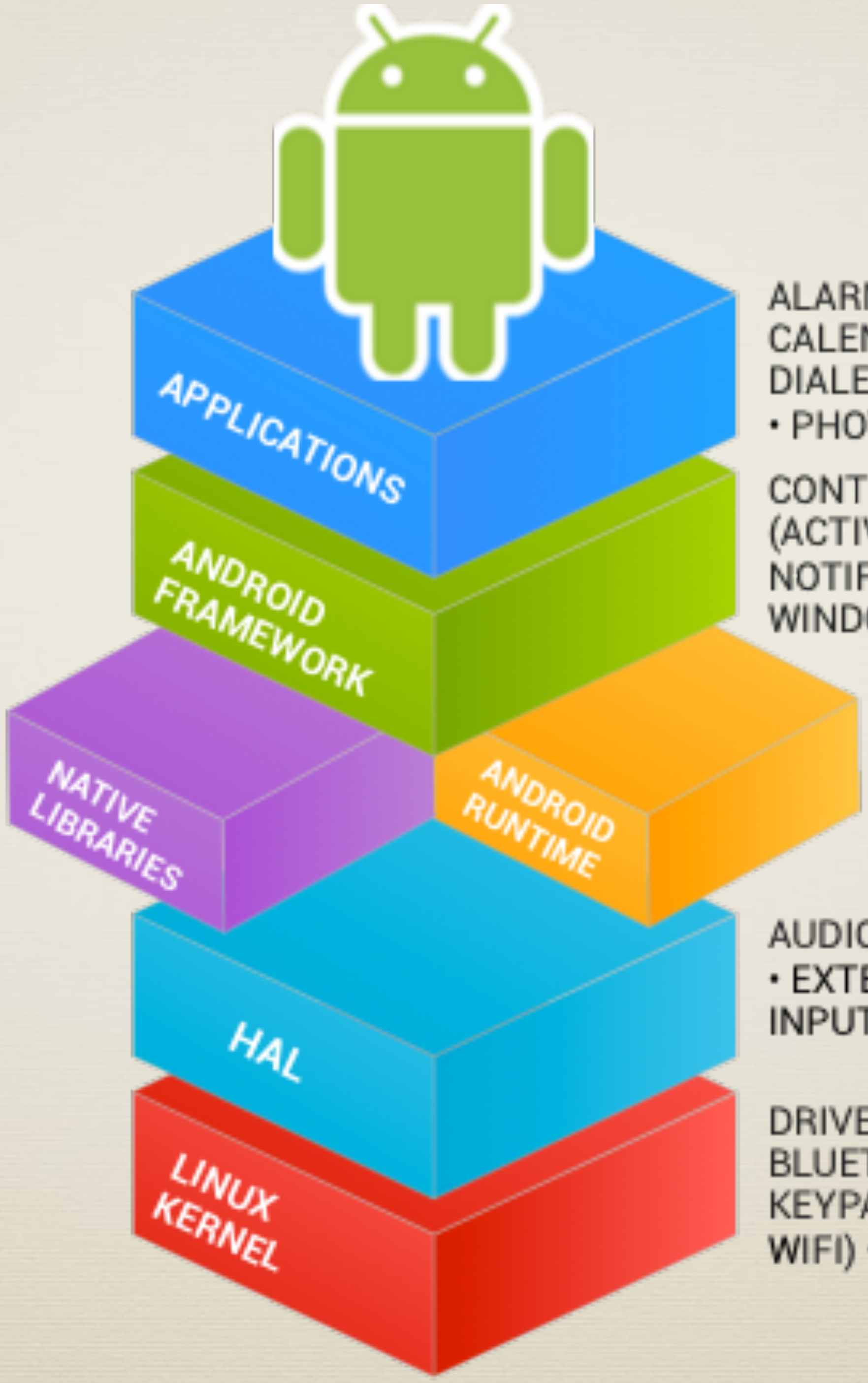
Android Boot Process

by mzlogin

GitHub: https://github.com/mzlogin

Pre-knowledge



APPLICATIONS

ALARM • BROWSER • CALCULATOR • CALENDAR • CAMERA • CLOCK • CONTACTS • DIALER • EMAIL • HOME • IM • MEDIA PLAYER • PHOTO ALBUM • SMS/MMS • VOICE DIAL

ANDROID FRAMEWORK

CONTENT PROVIDERS • MANAGERS (ACTIVITY, LOCATION, PACKAGE, NOTIFICATION, RESOURCE, TELEPHONY, WINDOW) • VIEW SYSTEM

NATIVE LIBRARIES

AUDIO MANAGER • FREETYPE • LIBC • MEDIA FRAMEWORK • OPENGL/ES • SQLITE • SSL • SURFACE MANAGER • WEBKIT

ANDROID RUNTIME

CORE LIBRARIES • ART • DALVIK VM

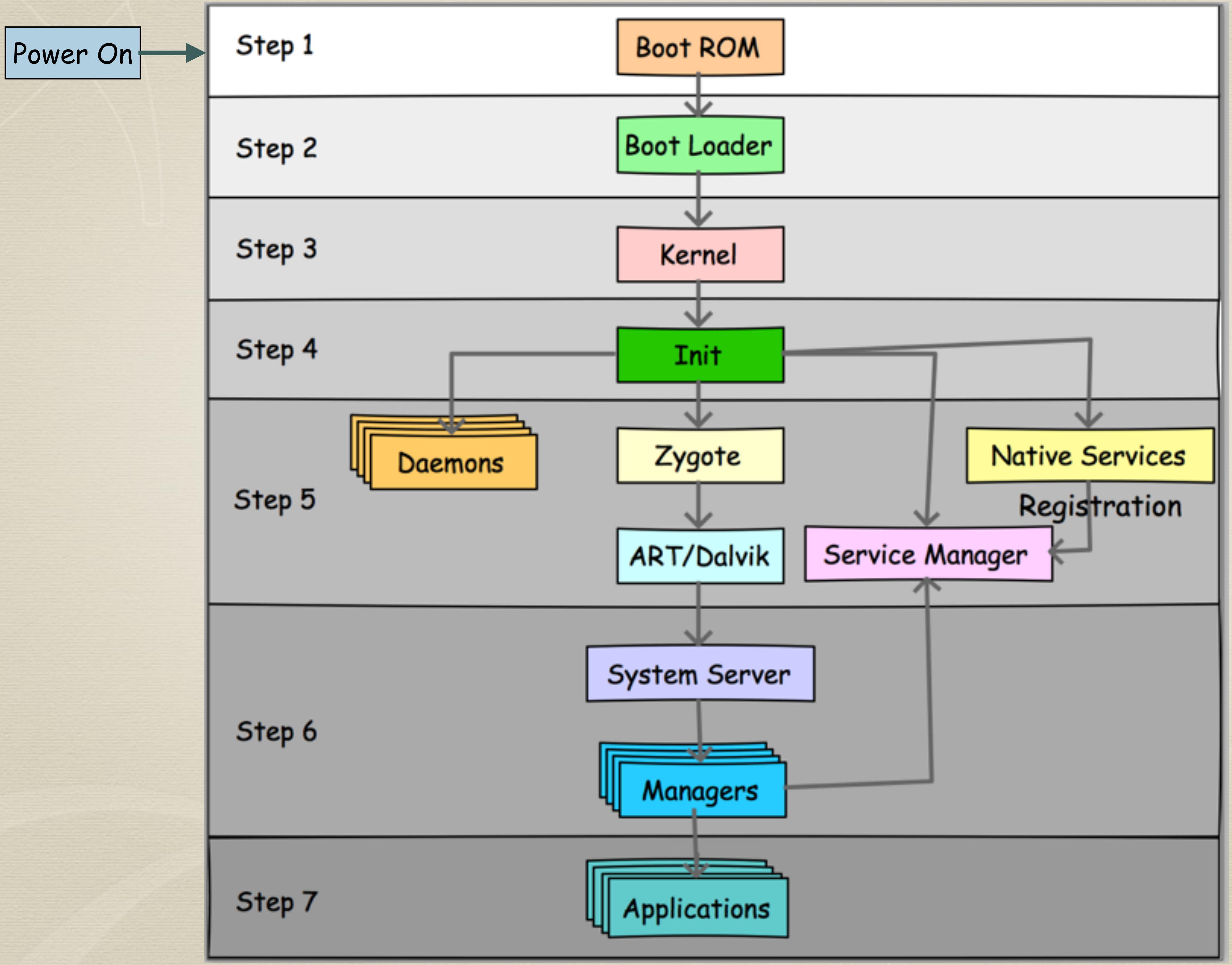
HAL

AUDIO • BLUETOOTH • CAMERA • DRM • EXTERNAL STORAGE • GRAPHICS • INPUT • MEDIA • SENSORS • TV

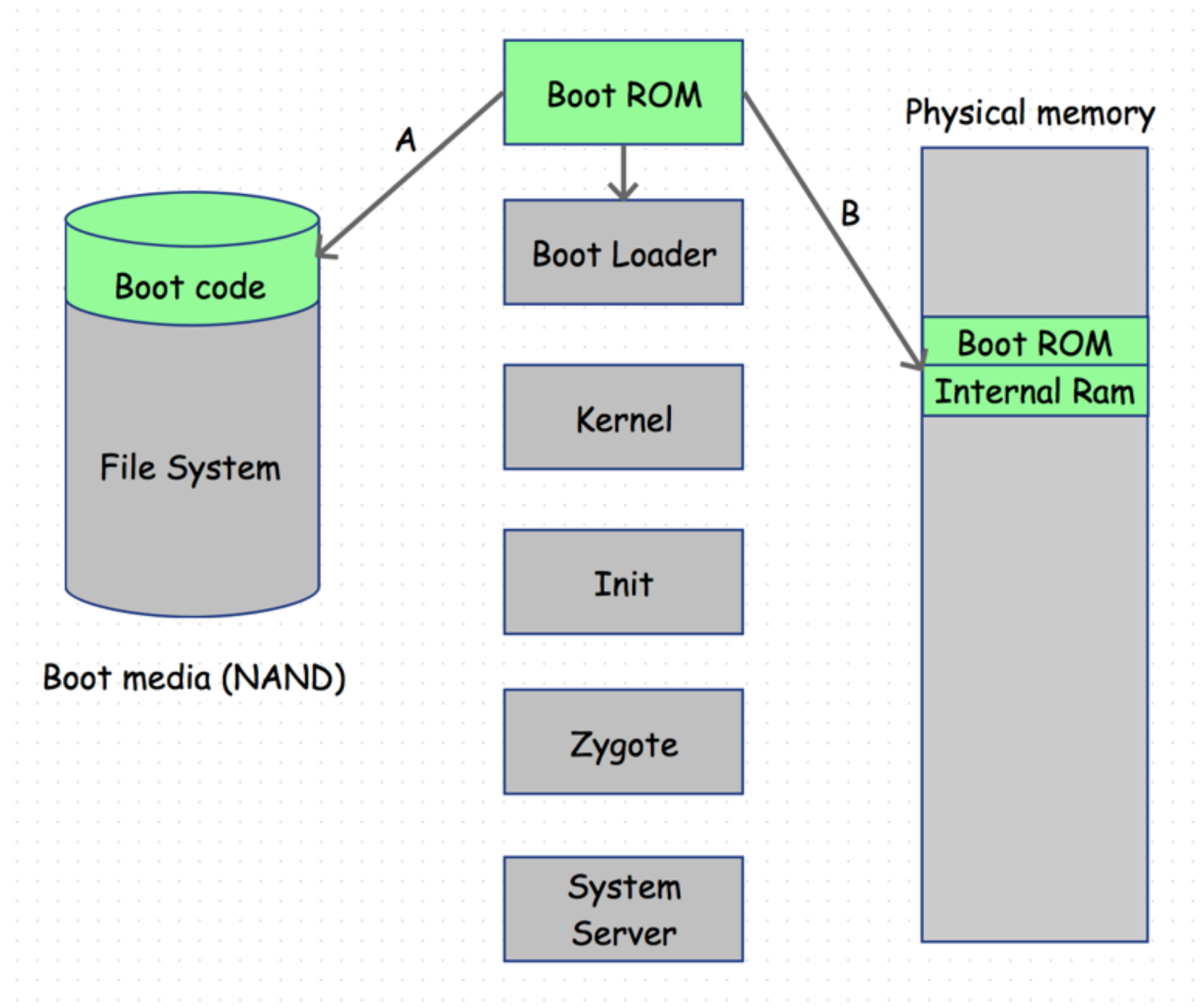
LINUX KERNEL

DRIVERS (AUDIO, BINDER (IPC), BLUETOOTH, CAMERA, DISPLAY, KEYPAD, SHARED MEMORY, USB, WIFI) • POWER MANAGEMENT

Summary


Power On
Step 1
Boot ROM
Step 2
Boot Loader
Step 3
Kernel
Step 4
Init
Step 5
Daemons
Zygote
Native Services
Registration
ART/Dalvik
Service Manager
Step 6
System Server
Managers
Step 7
Applications

Step 1. Power on and boot ROM code execution



Boot ROM

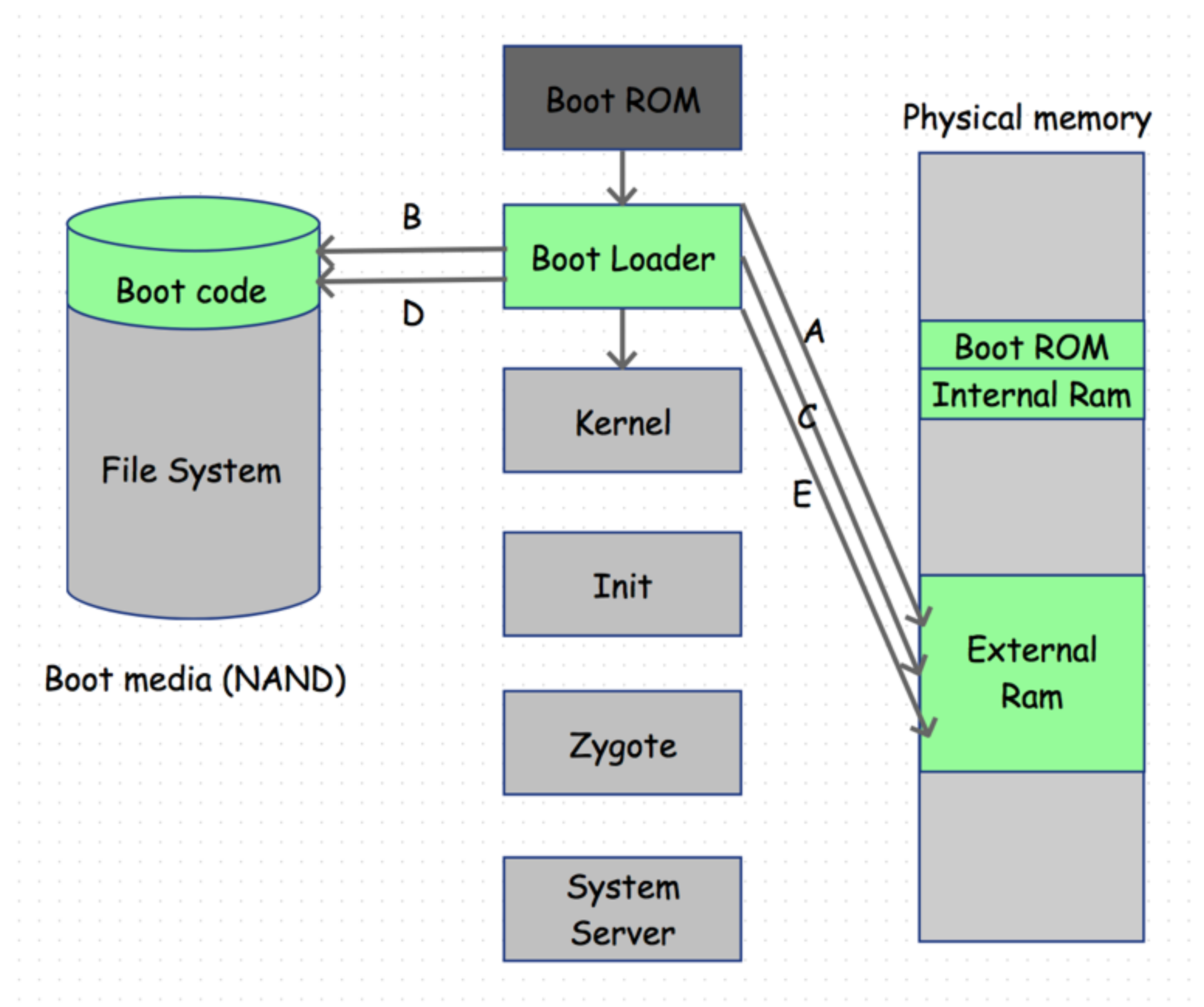
- a small piece of code that is hardwired in the CPU ASIC
- will execute when power supplies are stable

Sub-steps

A. Determine where to find the first stage of the boot loader.

B. Load the first stage boot loader to internal RAM, perform a jump and execution continues in the boot loader.

Step 2. The boot loader



Boot Loader

A program to set up initial memories and load the kernel to RAM.

Sub-steps

A. The first boot loader stage detect and set up external RAM.

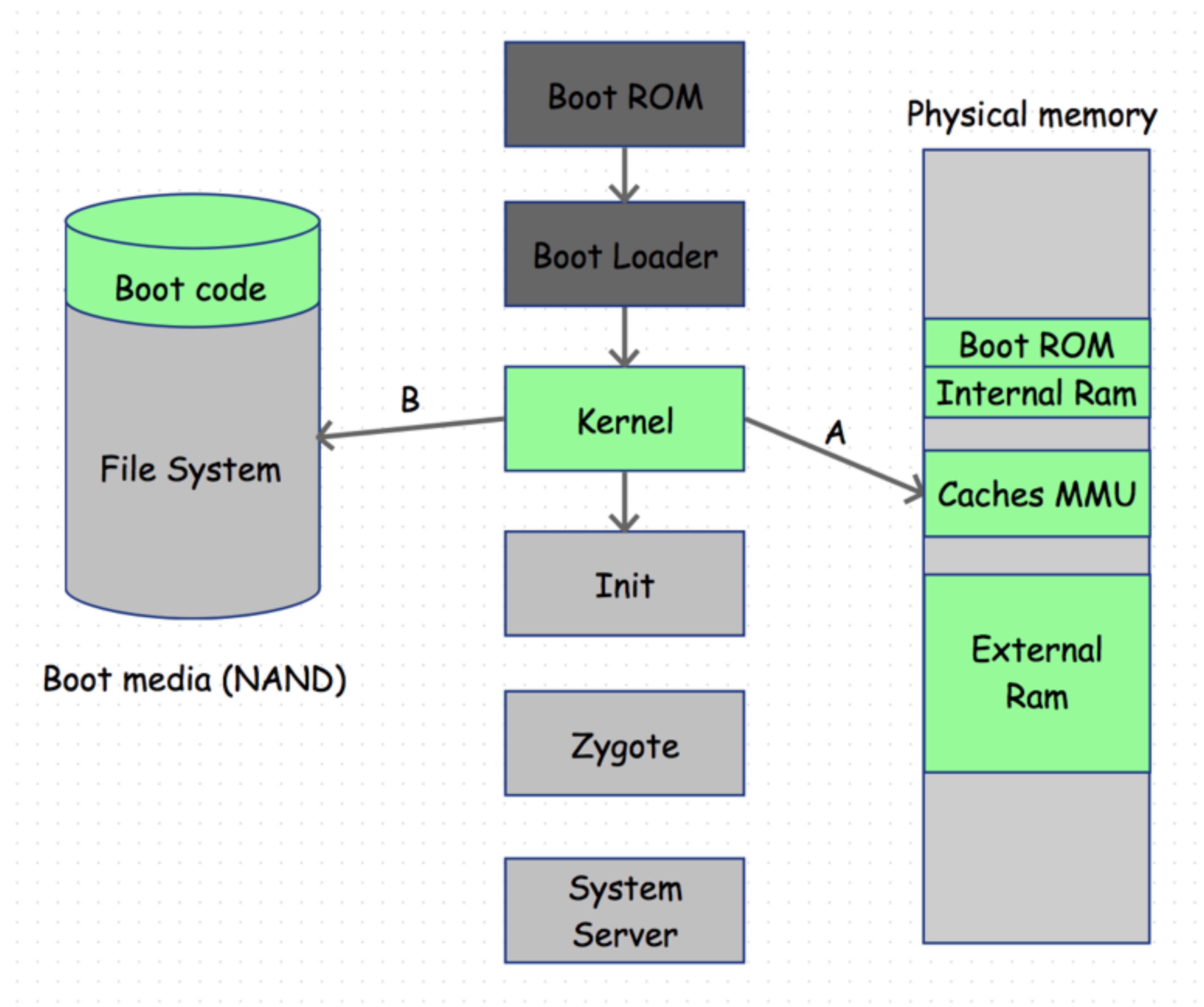
B. The first boot loader stage load the main boot loader to external RAM.

C. The second stage of the boot loader. (set up file systems, additional memory, network, low level memory protections and security options, etc.)

D. Look for a Linux kernel and place it in the RAM to boot.

E. Jump to the Linux kernel.

Step 3. The Linux kernel



Linux Kernel

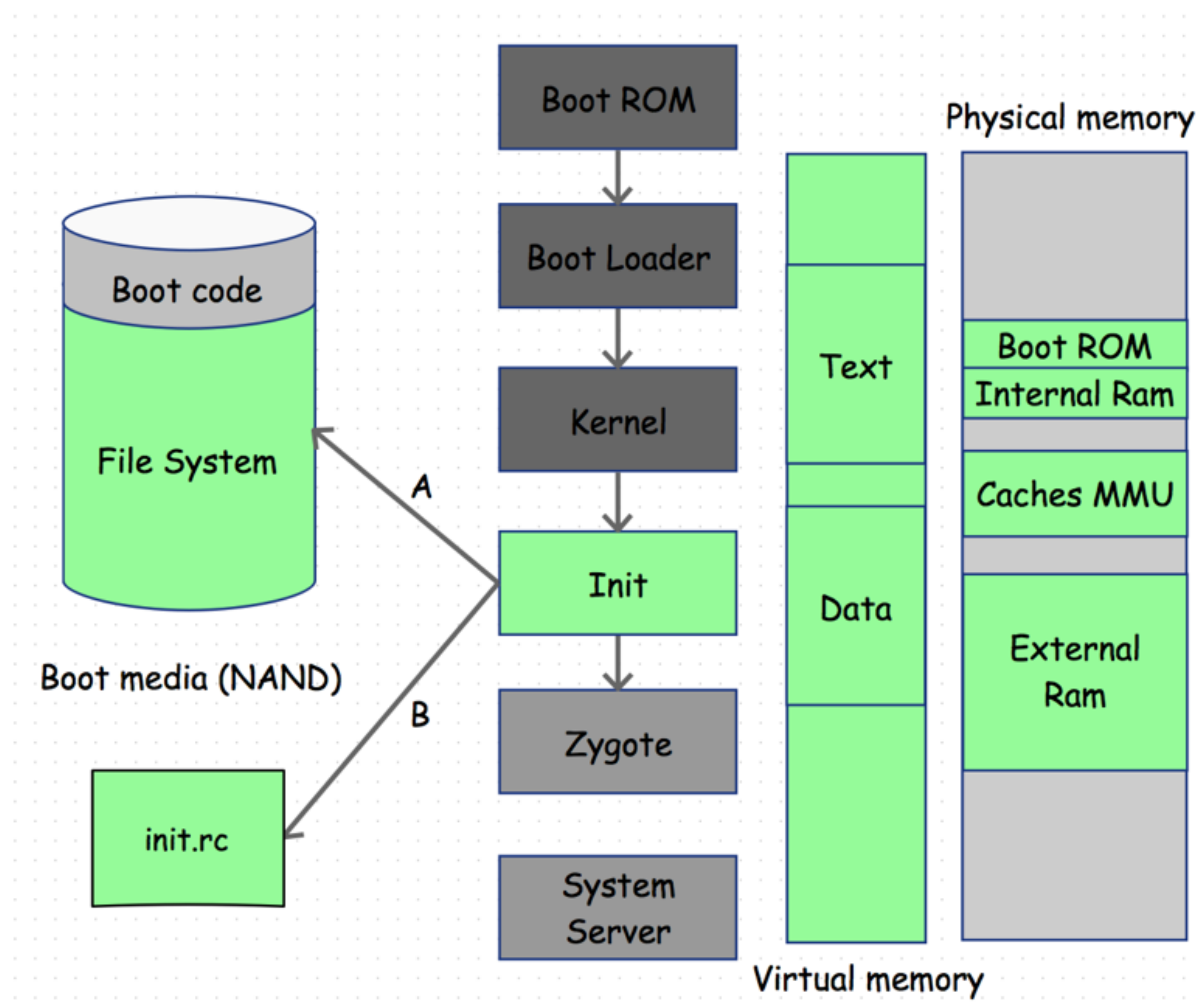
It will set up everything that is needed for the system to run. Initialize interrupt controllers, set up memory protections, caches and scheduling.

Sub-steps

A. Initialize the memory management units and caches, then the system will be able to use virtual memory and launch user space processes.

B. Launch init process as the initial user space process.

Step 4. The init process



Init Process

Grandmother of all processes.

Sub-steps

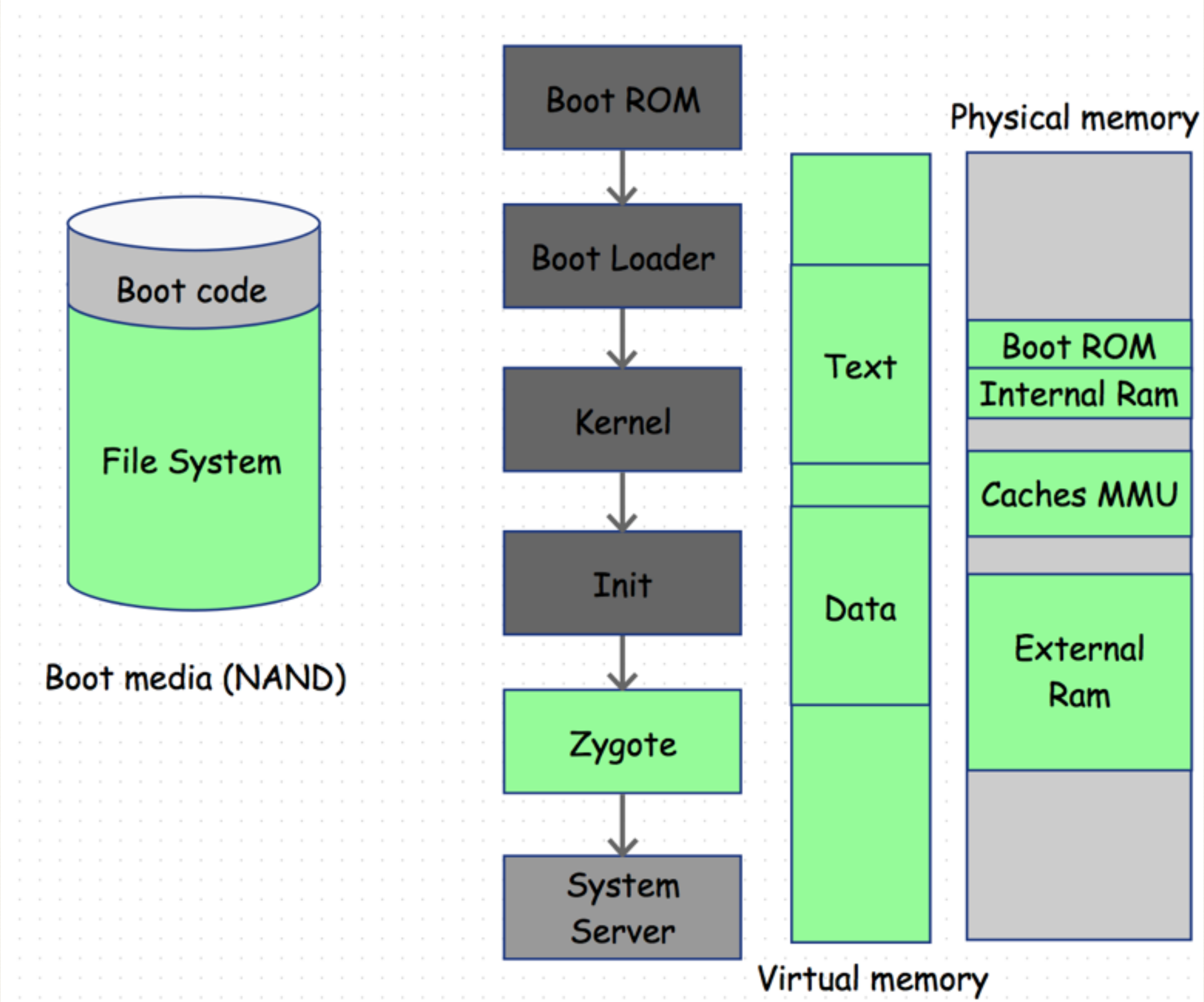
A. Mount directories like /sys, /dev, /proc, find init.rc file

B. Parse init.rc and run tasks in it

- global environment variables
- mount partitions
- directories permissions
- start services (daemons, native services, service manager, zygote, bootanim, ...)
- ...

At this stage you can see boot animation on device screen.

Step 5. Zygote and Dalvik/ART



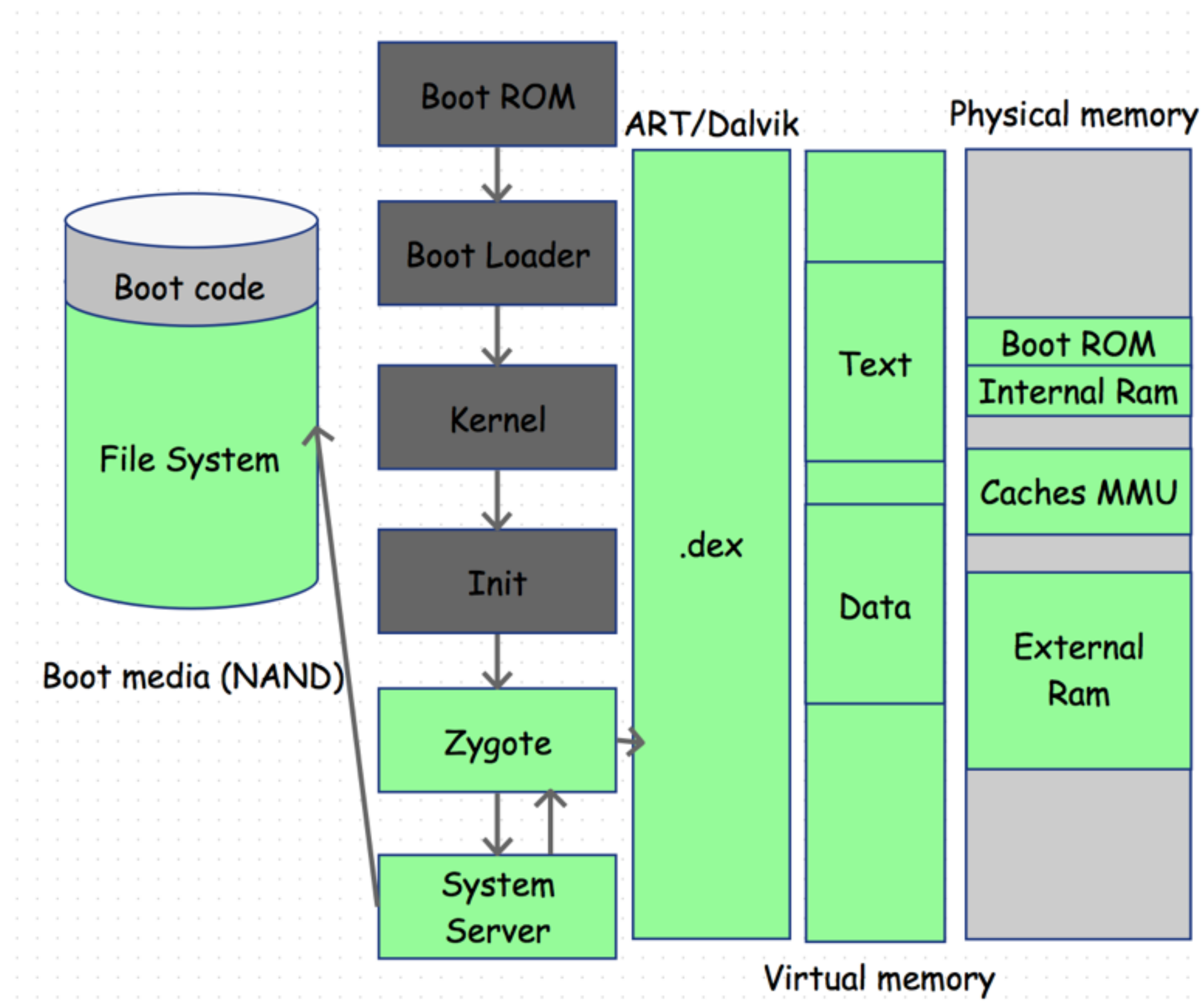
Zygote

All VM processes are forked from zygote.

Sub-steps

- registerZygoteSocket()
- preload()
  - preloadClasses() — preloaded-classes
  - preloadResources() — /system/framework/framework-res.apk
- startSystemServer()
- runSelectLoopMode()

Step 6. The system server



System Server

- First java component to run in the system
- It will start all the Android services.
  - Power Manager
  - Activity Manager
  - Telephony Registry
  - Scheduling Policy
  - Package Manager
  - Account Manager
  - ...

Step 7. Boot completed

- broadcast BOOT_COMPLETED
- start home activity

Extra knowledge: Tools for analyzing Android Bootup

- logcat

```
adb logcat -d -b events | grep boot

adb logcat -d | grep preload
```

- bootchart

References

- IN DEPTH : ANDROID BOOT SEQUENCE / PROCESS
- The Android boot process from power on
- What is the difference between a Bootrom vs bootloader on ARM systems
- Tools for analyzing Android Bootup
- Internal RAM
- What Is Internal RAM?

Q & A

Thanks