actor

FreeBSD Startup/Initialization

Turns system on
& motherboard
initializes BIOS → BIOS

BIOS starts in 16-bit real-mode

Runs POST
(Power On Self test)
& initializes hardware

Loads MBR (boot0)
from disk
(/boot/boot0) into
memory

Execution passed → MBR (boot0)

Loads First Slice of
FreeBSD OS
(/boot/boot1) from
disk into memory

First slice of OS contains boot1

Execution passed → boot1

Load boot2 & BTX
from disk (/boot/boot)
into memory

Enable access to
>1MB of memory

Execution passed → BTX Server

Sets up & switches to
32-bit protected env

Execution passed → boot2

Initializes
`struct bootinfo`

bootinfo contains info such as
kernel location, memory avail, etc

Loads /boot/loader
into memory
via `load()`

alt [If user presses key within 10s]

Displays prompt for user to customize boot settings

Actor inputs boot settings

Execution passed → loader

Loads kernel
into memory

Execution passed → kernel

Various cpu-specific
initializations
via `init386()`

Various system
initializations via
`mi_startup()`
(ie. scheduler)

Starts PID 1 (init)
via `create_init()` → init

init and everything to its right are programs in user space.
These programs likely make syscalls to the kernel.
To avoid overcomplicating the diagram, these calls won't be shown.

Forks process → rc

Run-level programs are
automatically started
that are defined
in /etc/rc

Return exit code

Other initializations
(ie. updates security
level to 1 or higher, etc)

FreeBSD Is Fully Initialized -- Usage Below

Forks process → getty

Forks process → login

loop [until user provides valid username & pass]

Requests user login

Provides username & password

Forks a shell
for the user → shell

Return shell for the user

loop [until shutdown with root privileges]

Command entered

Process command
(possibly making syscalls
to kernel, forking new
processes, running other
programs, writing output,
etc)

Return exit code

FreeBSD Shutdown

Shuts down system (ie. shutdown or poweroff)

Must be called with
root privileges
(ie. sudo shutdown)

Notifies of shutdown

Forks process → rc

cleanly
terminate processes
defined in
/etc/rc.shutdown

Return exit code

Terminates all left-over processes
(via TERM signal)

Returns exit code

Terminates all left-over processes
(via TERM signal)

Returns exit code

Terminates all left-over processes
(via TERM signal)

30 second timeout

Kills all left-over processes
(via KILL signal)

Note that **all left-over** process will be sent
TERM and then KILL signals after a 30s timeout.
This example assumes that rc() didn't kill these processes.
It also assumes that TERM killed the shell & getty
successfully, while login required a KILL signal.

Returns exit code

Logs shut down info

syscall to shut down system

actor | BIOS | MBR (boot0) | boot1 | BTX Server | boot2 | loader | kernel | init | rc | getty | login | shell