

# Babystep1

From OSDev Wiki

## Code

The following code is the smallest possible example of booting code from a floppy.

Difficulty level



Beginner

### Babystep1: Your first boot sector

Tutorial

Previous

Next

Babystep2

```
; boot.asm
hang:
    jmp hang

    times 512-
```

The CPU starts in real mode and the BIOS loads this code at address 0000:7c00. The "times 512..." stuff is NASM's way of saying fill up 512 bytes with zeros. And partcopy is going to expect that (200 in Hex = 512 in Decimal). Change it and you'll see Partcopy choke.

Often, you will see a so-called boot signature (0xAA55) at the end. Older versions of BIOSes looked for this in order to identify a boot sector on a disk. It is evidently unnecessary nowadays. If it's needed, the last line would be replaced with (or some version of it)

```
; boot.asm
hang:
    jmp hang

    times 510-($-$$) db 0 ; 2 bytes less now
    db 0x55
    db 0xAA
```

But the thing I'd really like to point out is how once you've booted, and the cursor is happily blinking on a blank screen, you might notice two things. One is that the floppy motor will turn off and the other is that you can press Ctrl-Alt-Del to reboot. The point is that interrupts (such as INT 0x09) are still being generated.

For kicks try clearing the interrupts flag:

```
;boot.asm
cli
hang:
    jmp hang

times 510-($-$$) db 0
db 0x55
db 0xAA
```

You may notice that the floppy motor doesn't turn off and you can't reboot with Ctrl-Alt-Del.

If you try to reduce this even more by removing the loop and merely pad out the sector with zeros, the BIOS will have something to say about it. On my machine, it was "Operating System Not Found". I have yet to try filling the sector with zeros except for adding a boot signature.

Not exactly something you would show your girlfriend, but I wanted to show just what the bare minimum is before I elaborate. Unless I'm irritating anyone, in which case I'll desist.

## Creating disk image

The code is assembled in NASM and copied to floppy using partcopy, dd, or debug. Then you simply boot from the floppy.

### Windows

```
nasmw boot.asm -f bin -o boot.bin
partcopy boot.bin 0 200 -f0
OR
debug boot.bin
-W 100 0 0 1
-Q
```

### Unix

```
nasm boot.asm -f bin -o boot.bin
dd if=boot.bin of=/dev/fd0
```

## References

- Instruction Set from the horse's mouth: [1] (<http://www.intel.com/design/pentiumII/documentation.htm>)
- Easier to read: [2] (<http://www.baldwin.cx/386htm/toc.htm>)
- NASM assembler - docs incl instruction set: [3] (<http://sourceforge.net/projects/nasm>)
- Partcopy - download pcopy02.zip (new link): [4] (<http://www.osdever.net/downloads.php>)
- Interrupts by number: [5] (<http://www.osdever.net/downloads.php>)

- Randall Hyde's look into the bowels of the PC: [6] (<http://webster.cs.ucr.edu/>)

Retrieved from "<http://wiki.osdev.org/index.php?title=Babystep1&oldid=14383>"

Categories:      Level 1 Tutorials | Babystep | Bootloaders

---

- This page was last modified on 23 January 2013, at 16:03.
- This page has been accessed 92,243 times.