IoT: Client Devices

Executable and Linking Format (ELF), a Quick Intro

What is it?

Various OS use various program formats

- Mach-o, PE, ELF are the big ones today

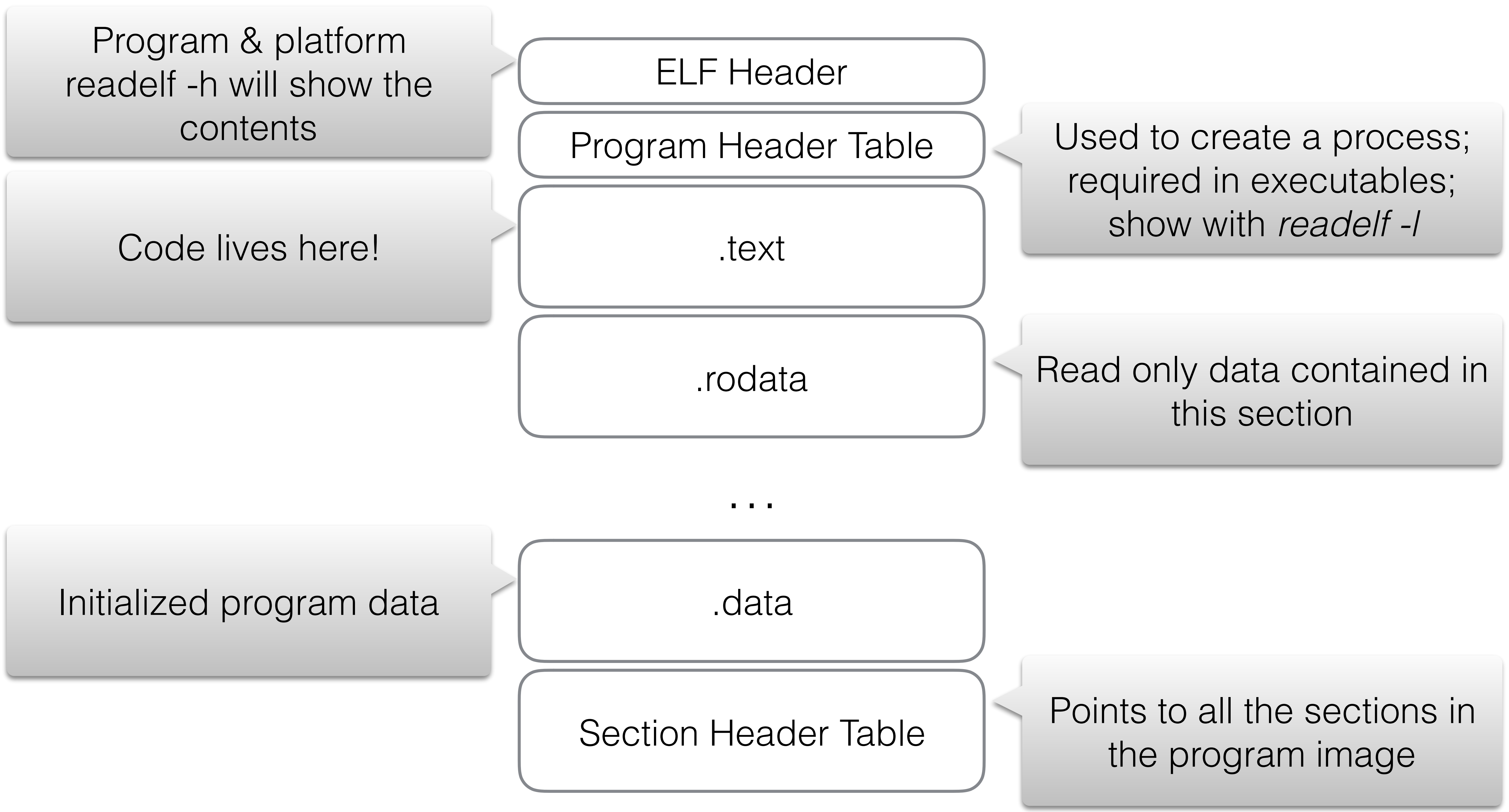
Ways to package programs

- Data, code, libraries, exported functions, etc.

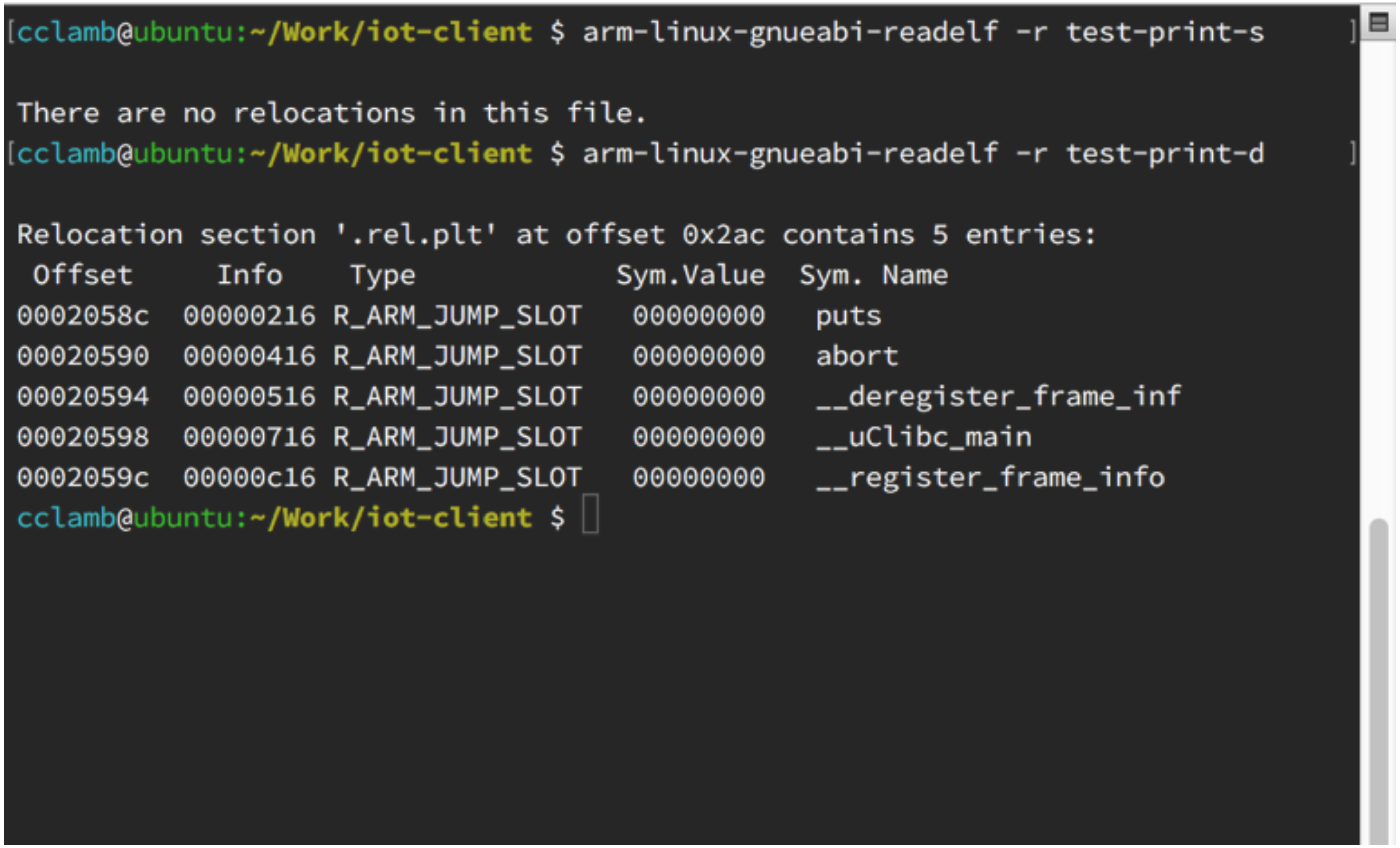
Used for different things

- Object (.o) files, libraries (.so and .a files), executables, core dumps

Organization



Program & platform readelf -h will show the contents
ELF Header
Program Header Table
Used to create a process; required in executables; show with *readelf -l*
Code lives here!
.text
.rodata
Read only data contained in this section
. . .
Initialized program data
.data
Section Header Table
Points to all the sections in the program image



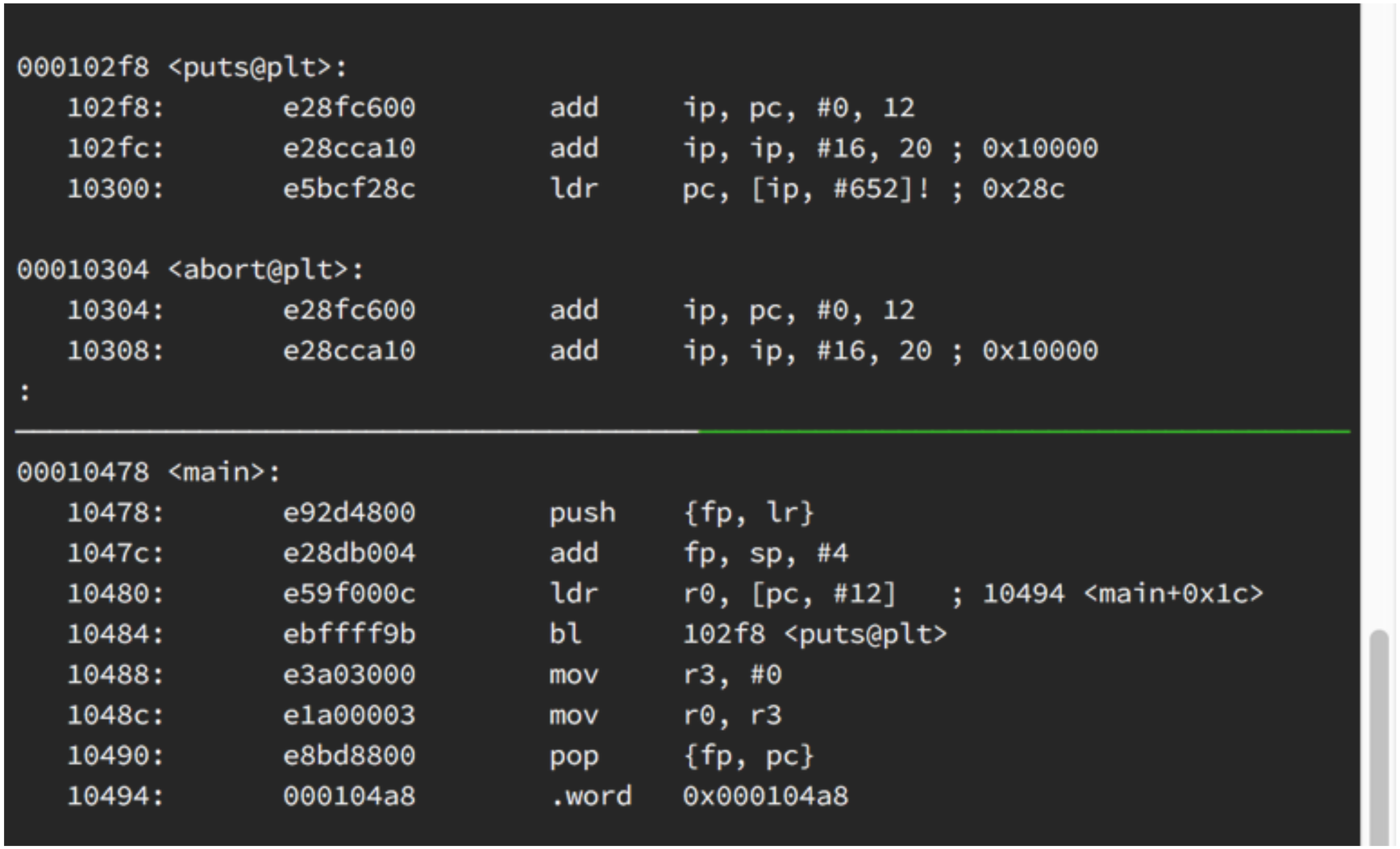
```
cclamb@ubuntu:~/Work/iot-client $ arm-linux-gnueabi-readelf -r test-print-s

There are no relocations in this file.
cclamb@ubuntu:~/Work/iot-client $ arm-linux-gnueabi-readelf -r test-print-d

Relocation section '.rel.plt' at offset 0x2ac contains 5 entries:
 Offset     Info    Type            Sym.Value  Sym. Name
0002058c  00000216 R_ARM_JUMP_SLOT   00000000   puts
00020590  00000416 R_ARM_JUMP_SLOT   00000000   abort
00020594  00000516 R_ARM_JUMP_SLOT   00000000   __deregister_frame_inf
00020598  00000716 R_ARM_JUMP_SLOT   00000000   __uClibc_main
0002059c  00000c16 R_ARM_JUMP_SLOT   00000000   __register_frame_info
cclamb@ubuntu:~/Work/iot-client $
```

Procedure Lookup Table

Static has no relocations, dynamic does



```
000102f8 <puts@plt>:
   102f8:       e28fc600        add     ip, pc, #0, 12
   102fc:       e28cca10        add     ip, ip, #16, 20 ; 0x10000
   10300:       e5bcf28c        ldr     pc, [ip, #652]! ; 0x28c

00010304 <abort@plt>:
   10304:       e28fc600        add     ip, pc, #0, 12
   10308:       e28cca10        add     ip, ip, #16, 20 ; 0x10000
:

00010478 <main>:
   10478:       e92d4800        push    {fp, lr}
   1047c:       e28db004        add     fp, sp, #4
   10480:       e59f000c        ldr     r0, [pc, #12]   ; 10494 <main+0x1c>
   10484:       ebffff9b        bl      102f8 <puts@plt>
   10488:       e3a03000        mov     r3, #0
   1048c:       e1a00003        mov     r0, r3
   10490:       e8bd8800        pop     {fp, pc}
   10494:       000104a8        .word   0x000104a8
```

How do we relocate?

objdump -S test-print-d