IoT: Client Devices

Linking & Loading

Linkers

CREATES EXECUTABLE IMAGES

- Libraries, executables, etc.

USES OBJECT FILES

- .o files; you can see these when you build (usually)
- By extension, static libraries too (.a files)

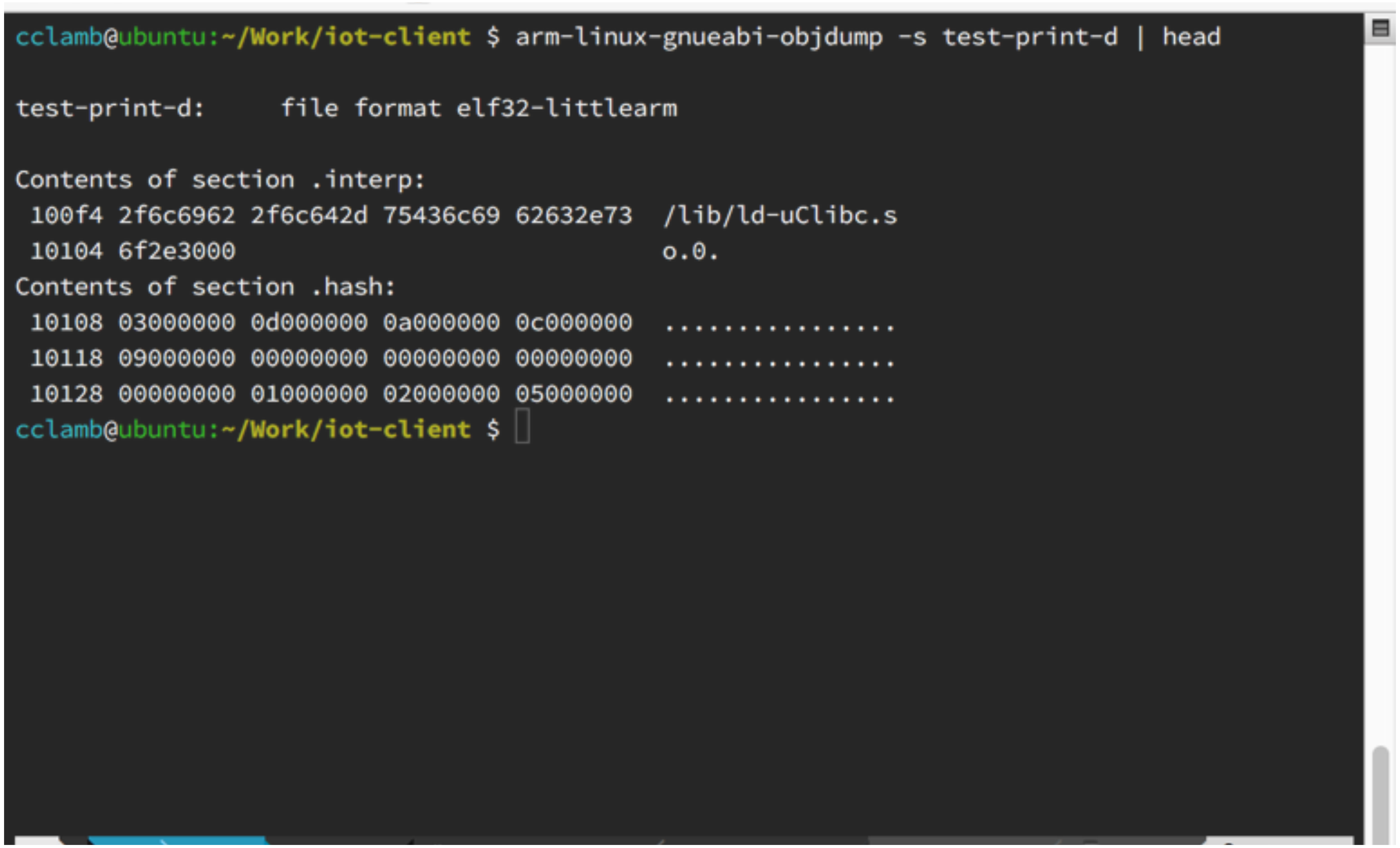
Loaders

BOOTLOADERS, EMBEDDED SYSTEMS

- Bootloaders are special loaders, load OS/Kernel
- Embedded systems frequently do not have loaders
- We’re using embedded linux though, which has one

LOADS PROGRAMS AND DYNAMIC LIBRARIES

- Loads programs into memory, starts execution (at _start)
- Sometimes uses a dynamic linker
- Executables use them



```
cclamb@ubuntu:~/Work/iot-client $ arm-linux-gnueabi-objdump -s test-print-d | head

test-print-d:     file format elf32-littlearm

Contents of section .interp:
 100f4 2f6c6962 2f6c642d 75436c69 62632e73  /lib/ld-uClibc.s
 10104 6f2e3000                             o.0.
Contents of section .hash:
 10108 03000000 0d000000 0a000000 0c000000  ................
 10118 09000000 00000000 00000000 00000000  ................
 10128 00000000 01000000 02000000 05000000  ................
cclamb@ubuntu:~/Work/iot-client $
```

ARM Dynamic Linker

Dynamic linker path is embedded in executable

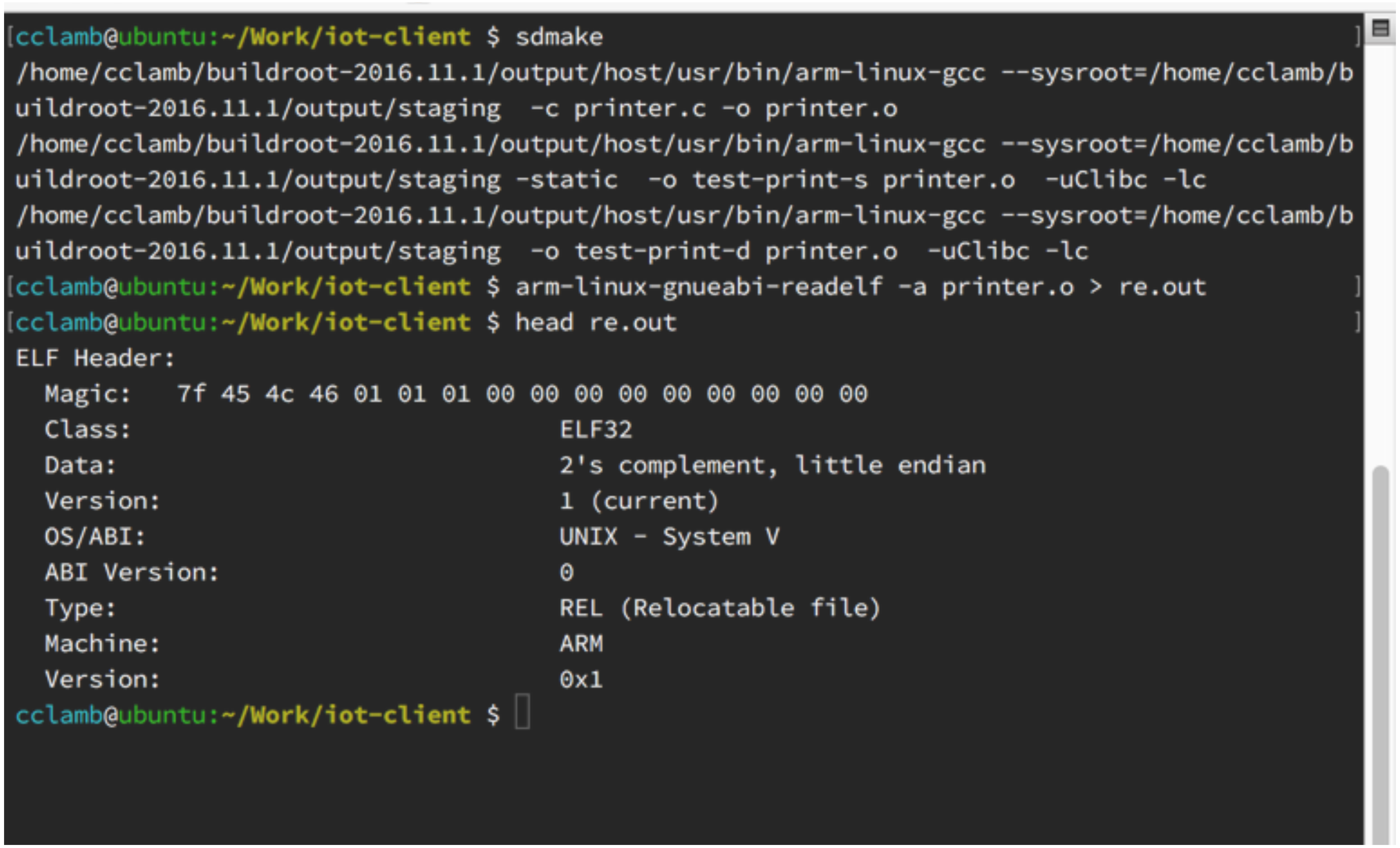
Object Files

Object files contain object code

- Relocatable instructions for a platform
- Not directly executable

Relocatability is important

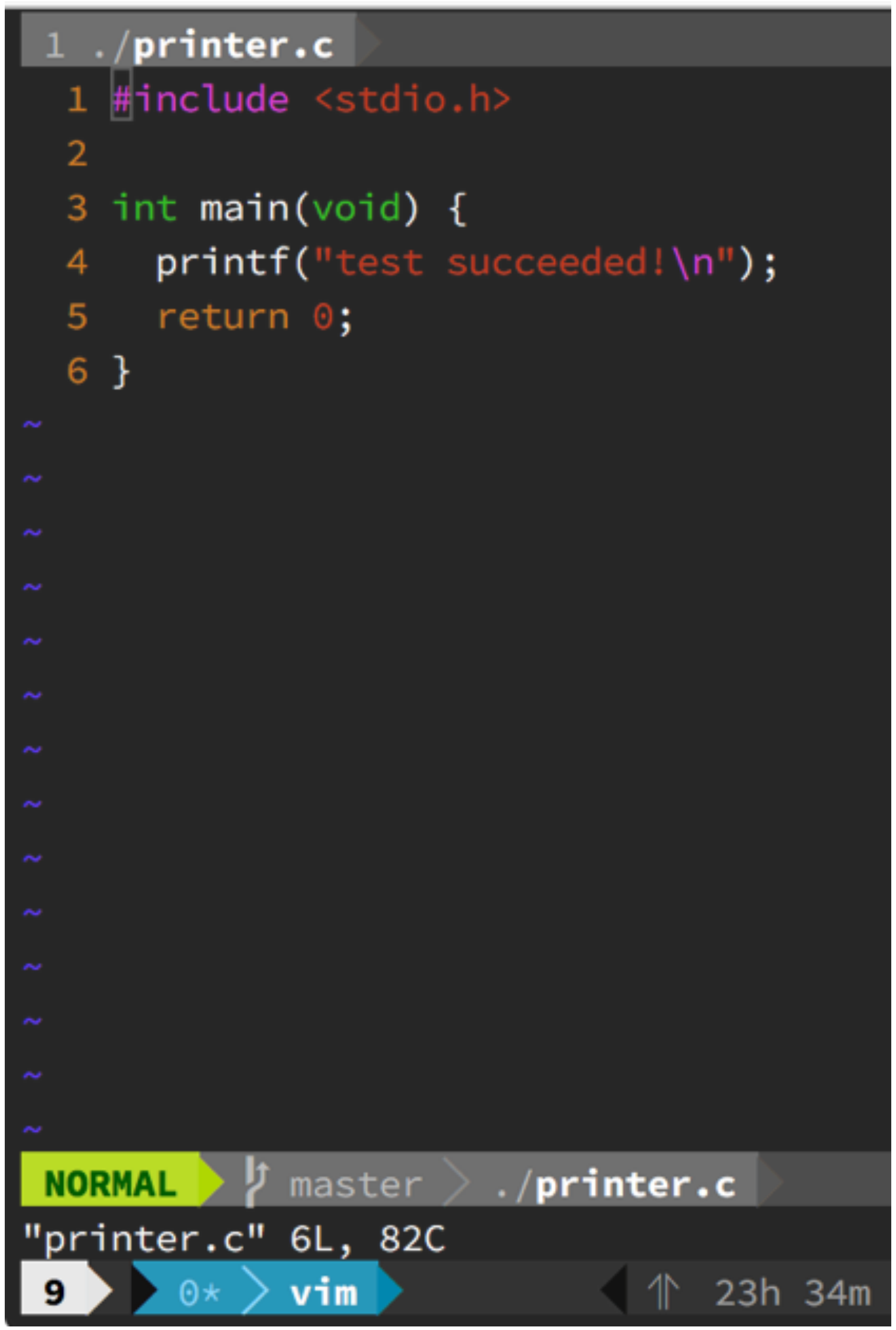
- The object code is inserted by the linker into a dynamic library or executable image
- Relocatability allows linker to place code arbitrarily (-ish)



```
cclamb@ubuntu:~/Work/iot-client $ sdmake
/home/cclamb/buildroot-2016.11.1/output/host/usr/bin/arm-linux-gcc --sysroot=/home/cclamb/b
uildroot-2016.11.1/output/staging  -c printer.c -o printer.o
/home/cclamb/buildroot-2016.11.1/output/host/usr/bin/arm-linux-gcc --sysroot=/home/cclamb/b
uildroot-2016.11.1/output/staging -static  -o test-print-s printer.o  -uClibc -lc
/home/cclamb/buildroot-2016.11.1/output/host/usr/bin/arm-linux-gcc --sysroot=/home/cclamb/b
uildroot-2016.11.1/output/staging  -o test-print-d printer.o  -uClibc -lc
cclamb@ubuntu:~/Work/iot-client $ arm-linux-gnueabi-readelf -a printer.o > re.out
cclamb@ubuntu:~/Work/iot-client $ head re.out
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                             ELF32
  Data:                              2's complement, little endian
  Version:                           1 (current)
  OS/ABI:                            UNIX - System V
  ABI Version:                       0
  Type:                              REL (Relocatable file)
  Machine:                           ARM
  Version:                           0x1
cclamb@ubuntu:~/Work/iot-client $
```

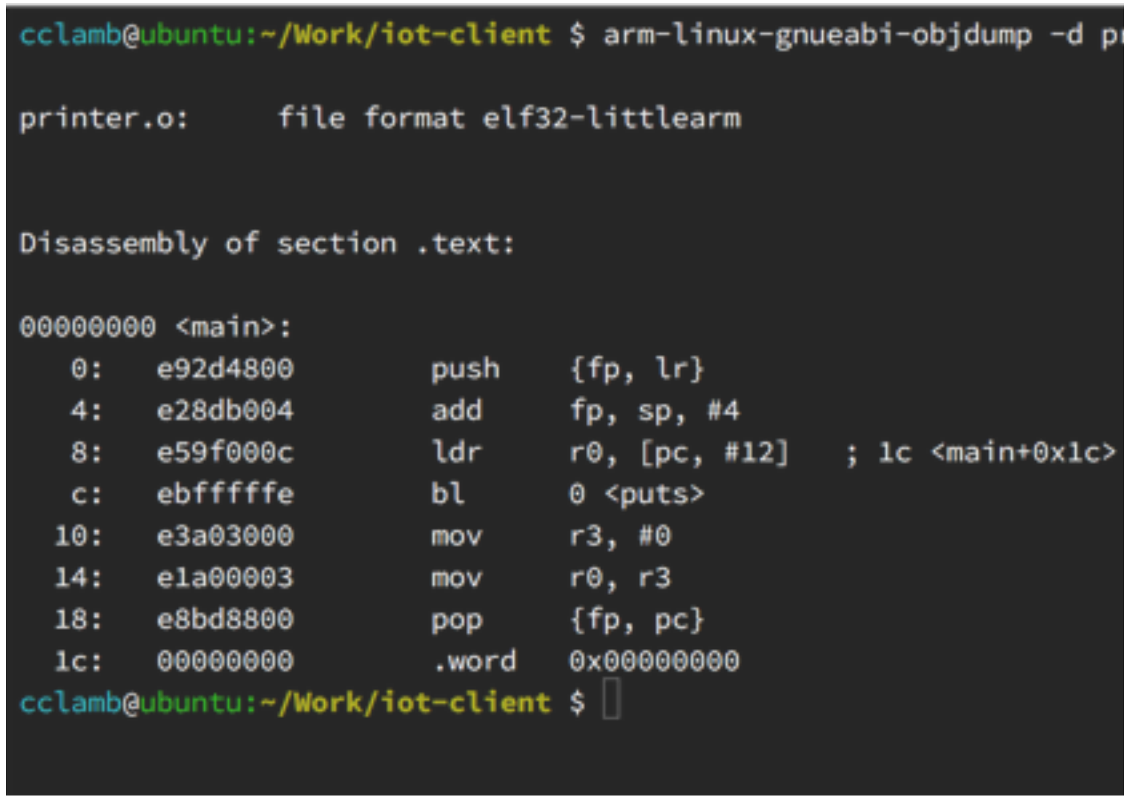
Object File Example

Using our old printer example



```
1 ./printer.c
#include <stdio.h>

int main(void) {
  printf("test succeeded!\n");
  return 0;
}
~
NORMAL  master  ./printer.c
"printer.c" 6L, 82C
9  0*  vim                         23h 34m
```

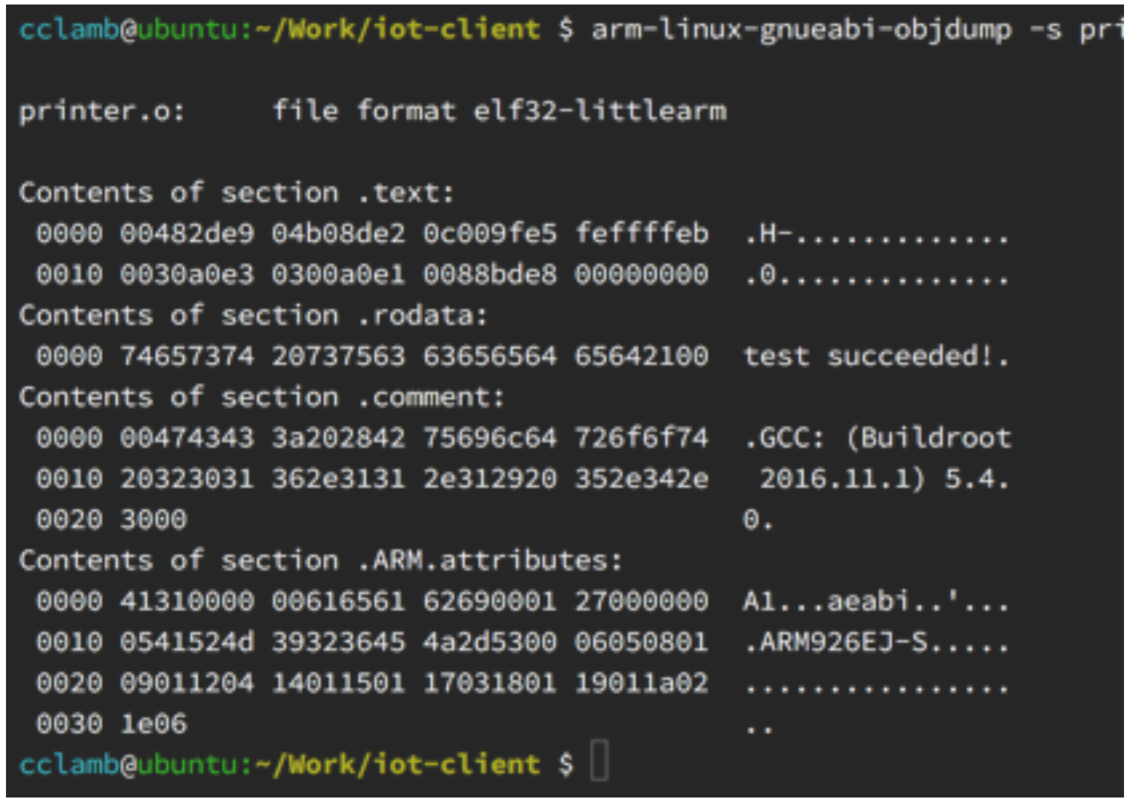


```
cclamb@ubuntu:~/Work/iot-client $ arm-linux-gnueabi-objdump -d pr

printer.o:     file format elf32-littlearm


Disassembly of section .text:

00000000 <main>:
   0:   e92d4800        push    {fp, lr}
   4:   e28db004        add     fp, sp, #4
   8:   e59f000c        ldr     r0, [pc, #12]   ; 1c <main+0x1c>
   c:   ebfffffe        bl      0 <puts>
  10:   e3a03000        mov     r3, #0
  14:   e1a00003        mov     r0, r3
  18:   e8bd8800        pop     {fp, pc}
  1c:   00000000        .word   0x00000000
cclamb@ubuntu:~/Work/iot-client $ 
```



```
cclamb@ubuntu:~/Work/iot-client $ arm-linux-gnueabi-objdump -s pri

printer.o:     file format elf32-littlearm

Contents of section .text:
 0000 00482de9 04b08de2 0c009fe5 feffffeb  .H-.............
 0010 0030a0e3 0300a0e1 0088bde8 00000000  .0..............
Contents of section .rodata:
 0000 74657374 20737563 63656564 65642100  test succeeded!.
Contents of section .comment:
 0000 00474343 3a202842 75696c64 726f6f74  .GCC: (Buildroot
 0010 20323031 362e3131 2e312920 352e342e   2016.11.1) 5.4.
 0020 3000                                 0.
Contents of section .ARM.attributes:
 0000 41310000 00616561 62690001 27000000  A1...aeabi..'...
 0010 0541524d 39323645 4a2d5300 06050801  .ARM926EJ-S.....
 0020 09011204 14011501 17031801 19011a02  ................
 0030 1e06                                 ..
cclamb@ubuntu:~/Work/iot-client $ 
```