Memory Safety: Attacks and Defense (Demos)

- Show vulnerable program
- Dissect program, objdump
- Load program using GDB
    - Basic use of GDB
- Three tasks in GDB:
    - Break program / Control-flow Hijacking / Shellcode injection
- Two defenses:
    - ASLR, Stack Canaries

Group question: find the vulnerable piece in this program

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void foo(char *str) {
  char buffer[100];
  strcpy(buffer, str);
}

int main(int argc, char **argv) {
  foo(argv[1]);
}
```

Exercise: how does this program get compiled?

Once we have the binary, what does it look like?

Enter `objdump / readelf`

```
objdump -D cpyarg
```

Disassemble

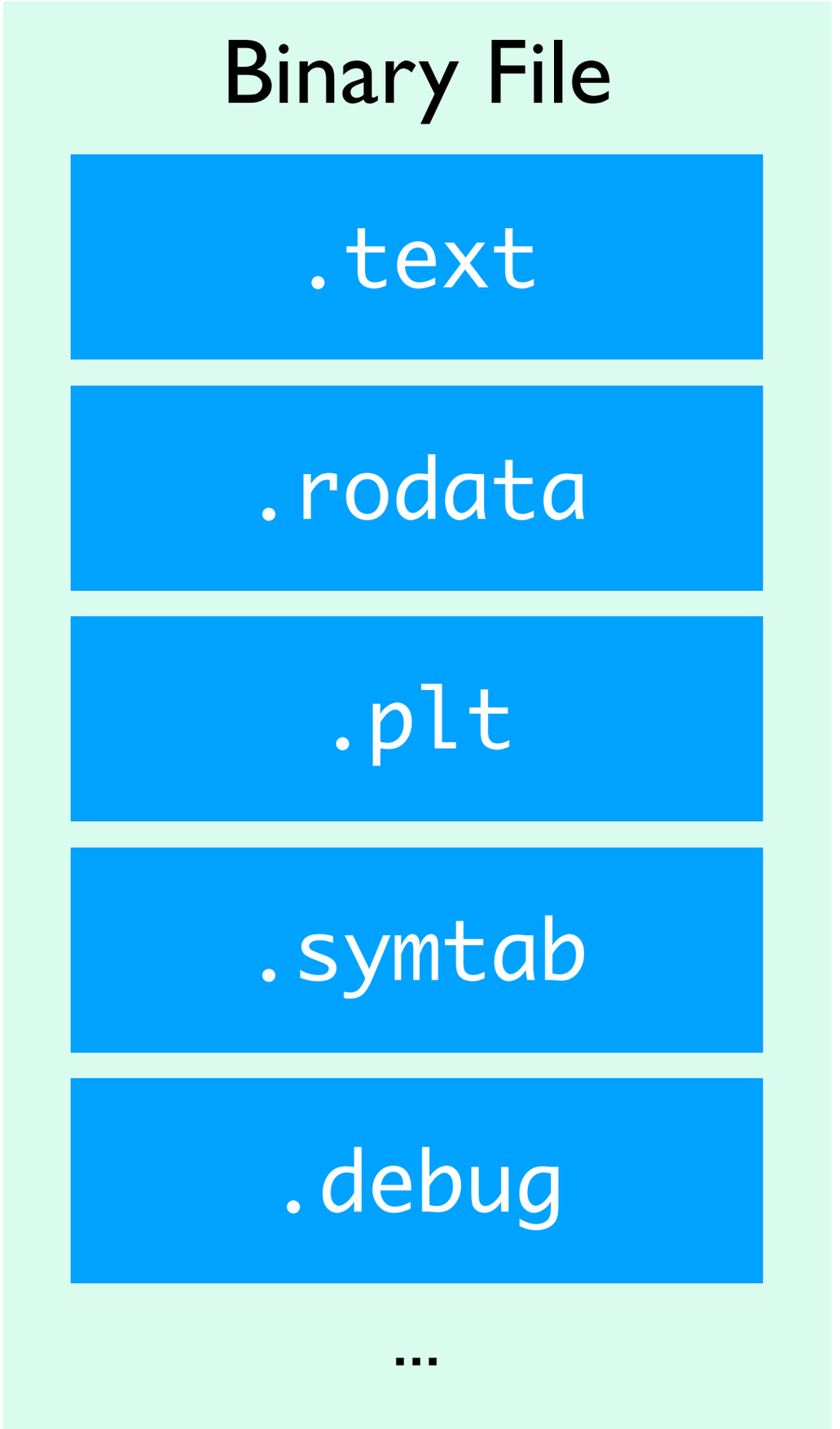
```
readelf -a cpyarg
```

Written in ELF

(Executable Linking Format)

Purpose of this format is to tell computer how to **set up** a binary

File composed of many **sections**



Written in ELF

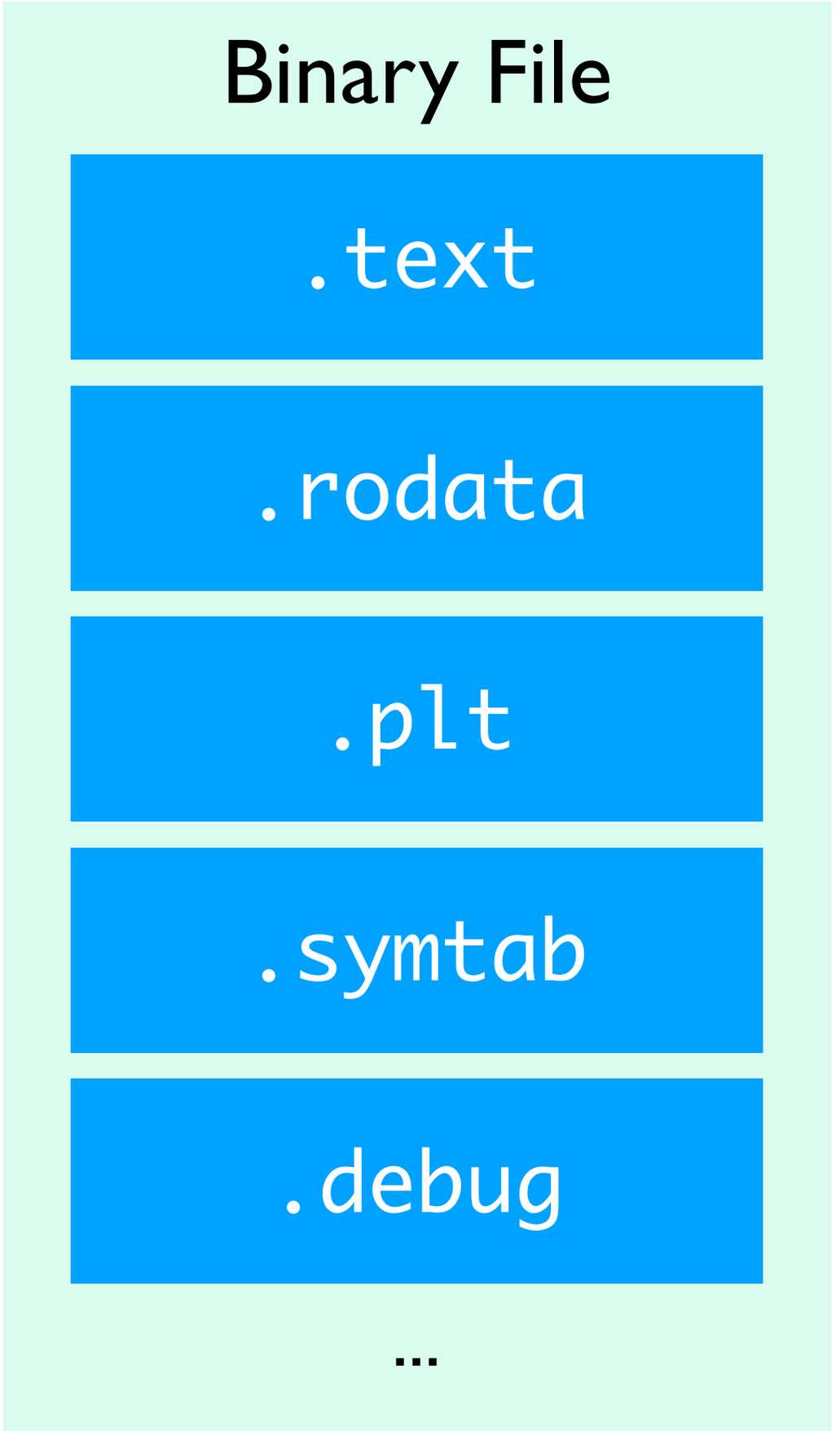
(Executable Linking Format)

Purpose of this format is to tell computer how to **set up** a binary

File composed of many **sections**

Kernel then **loads** these into memory

(Other things: dynamic linking, won't discuss here)



Binary File

.text

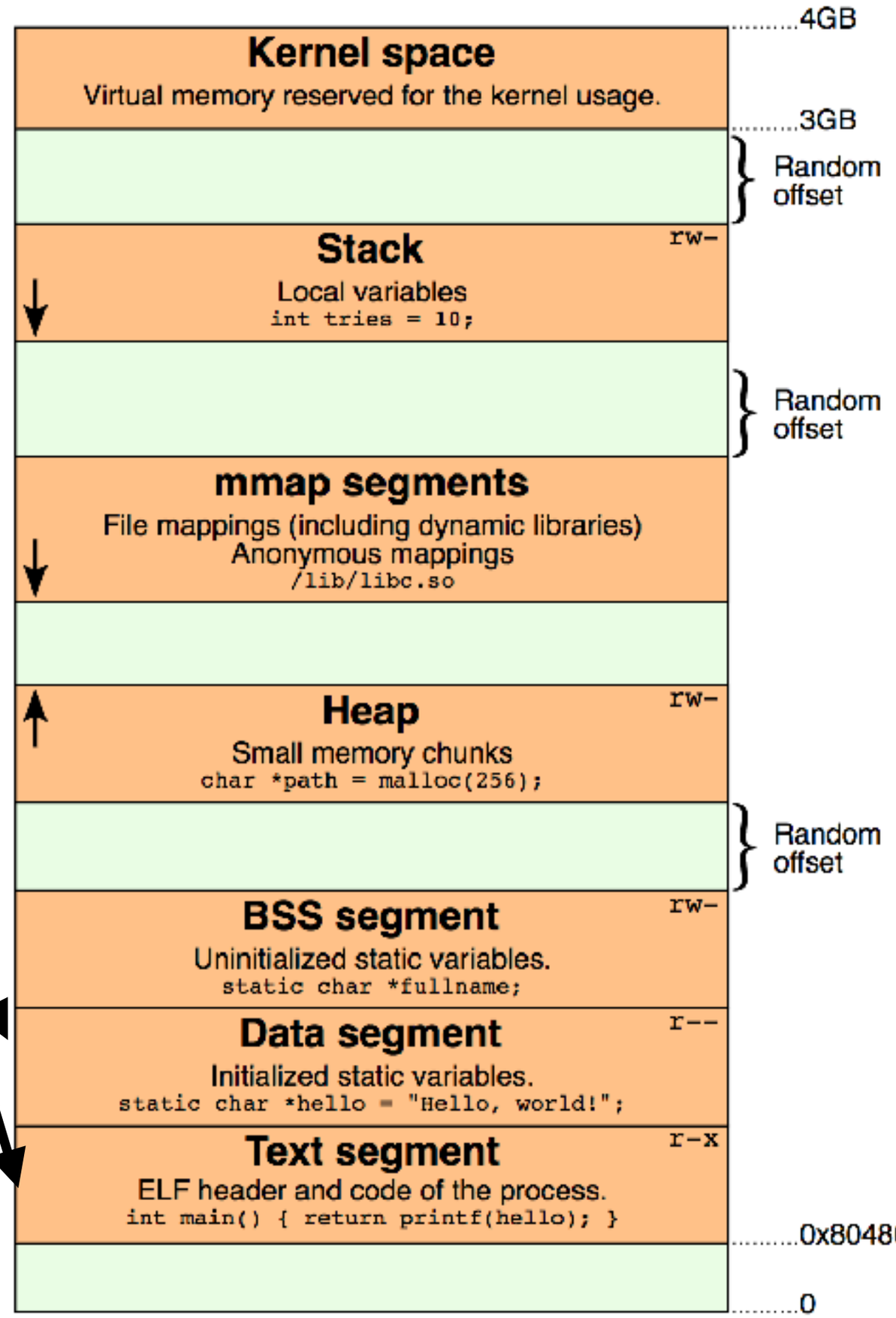
.rodata

.plt

.symtab

.debug

...

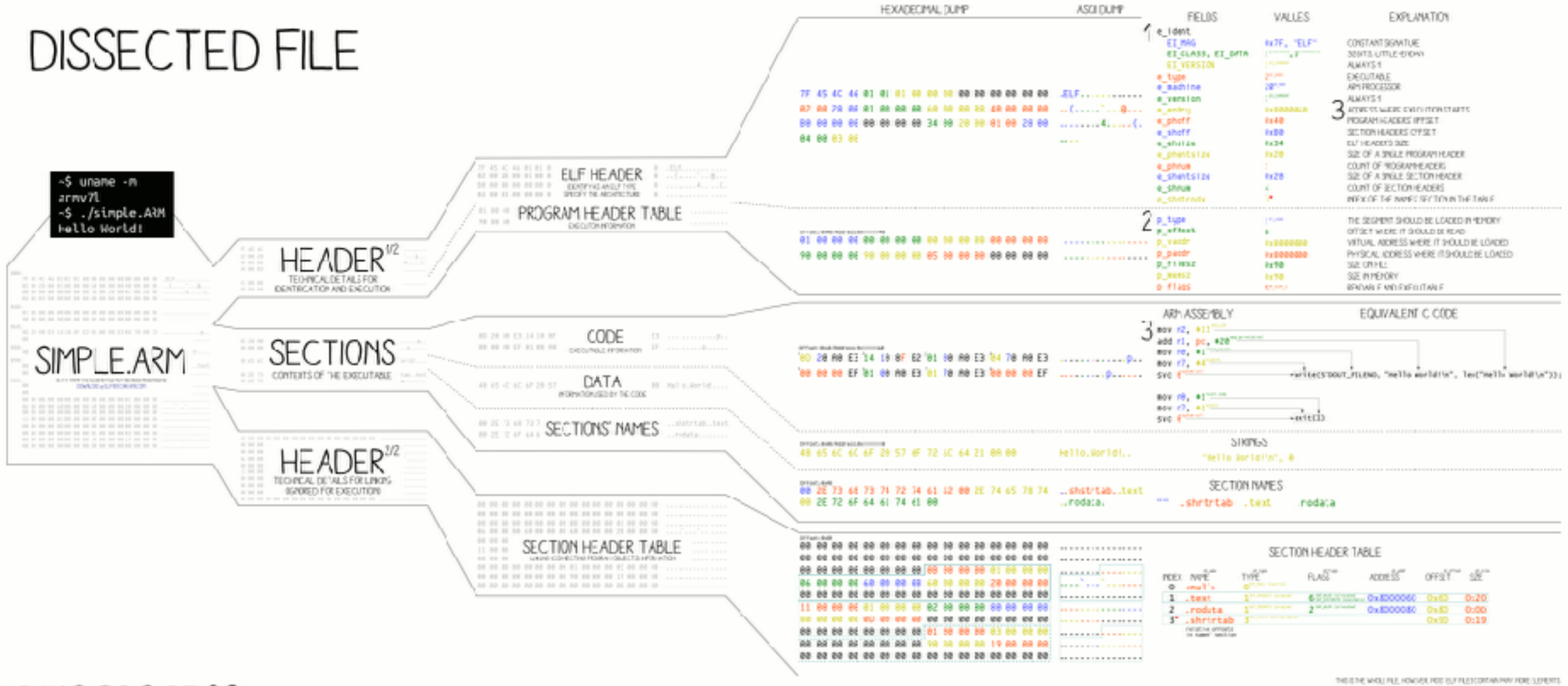


http://www.cirosantilli.com/elf-hello-world/

ELF[101] a Linux executable walkthrough

ANGE ALBERTINI
CORKAMI.COM

DISSECTED FILE



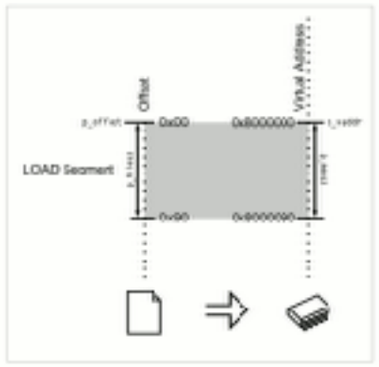
LOADING PROCESS

1 HEADER

THE ELF HEADER IS PARSED
THE PROGRAM HEADER IS PARSED
(SECTIONS ARE NOT USED)

2 MAPPING

THE FILE IS MAPPED IN MEMORY
ACCORDING TO ITS SEGMENT(S)



3 EXECUTION

ENTRY IS CALLED
SYSCALLS[2] ARE ACCESSED VIA:
- SYSCALL NUMBER IN THE R7 REGISTER
- CALLING INSTRUCTION SVC

TRIVIA

THE ELF WAS FIRST SPECIFIED BY U.S.L.[3] AND U.I.[4]
FOR UNIX SYSTEM V, IN 1989

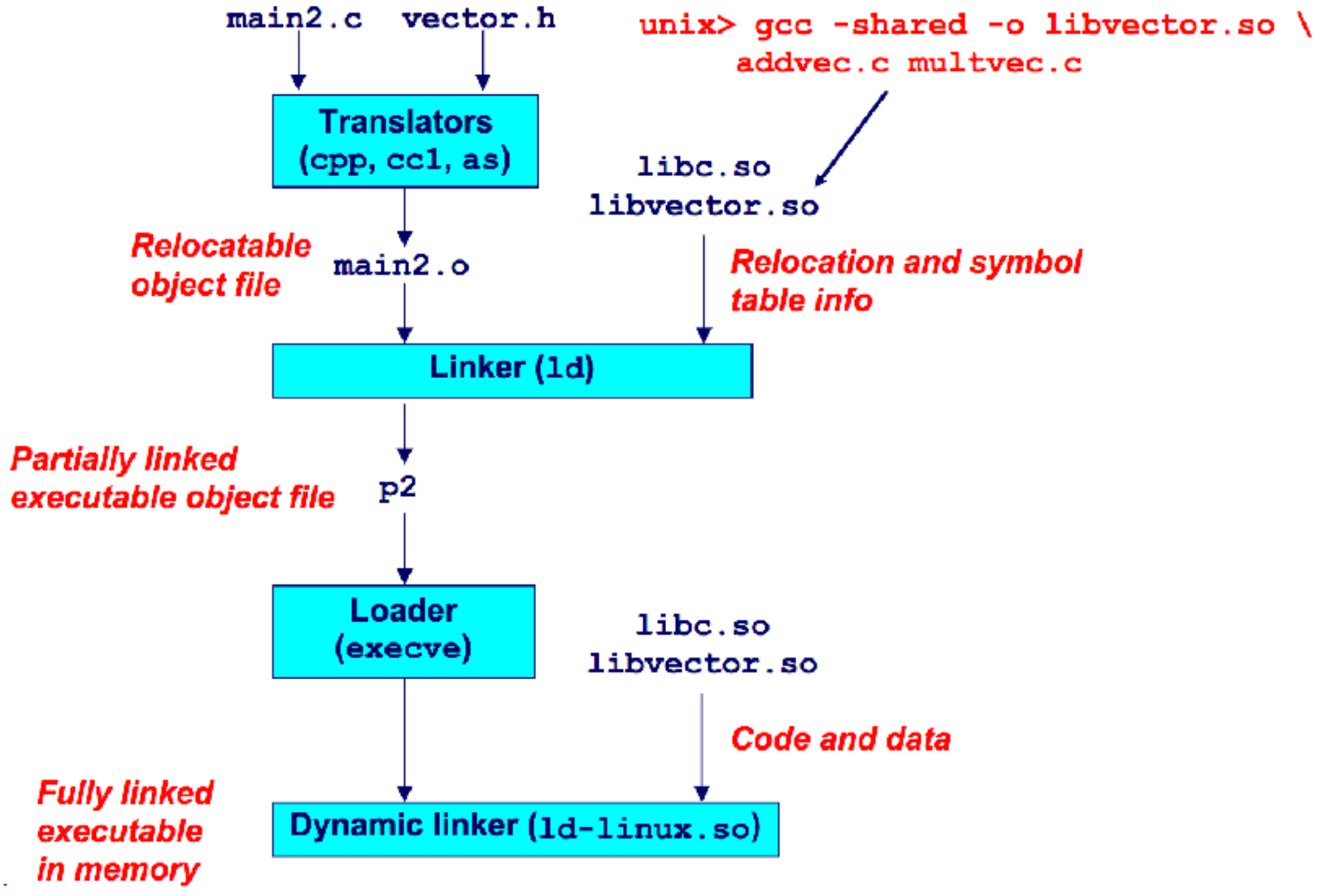
THE ELF IS USED, AMONG OTHERS, IN:
- LINUX, ANDROID, *BSD, SOLARIS, BEOS
- PSP, PLAYSTATION 2-4, DREAMCAST, GAMECUBE, WII
- VARIOUS OSES MADE BY SAMSUNG, ERICSSON, NOKIA,
- MICROCONTROLLERS FROM ATMEL, TEXAS INSTRUMENTS

VERSION 1.0[5]
2013/12/06

Why no code for functions from libc?

Answer: **dynamically** linked into the file

Upshot: dynamic linker "moves around" program to work



http://slideplayer.com/slide/8579139/

Poking around the program: GDB

```
i f
```

Show **info** about the current **frame**
(prev. frame, locals/args, %rbp/%rip)

```
i r
```

Show **info** about **reg**isters
(%rip, %rbp, %rsp, etc.)

```
x/<n> <addr>
```

E**x**amine <n> bytes of memory
starting at address <addr>

```
b <function>
s
```

Set a **b**reakpoint at <function>
step through execution (into calls)