CSE 330: Operating Systems

Class: 04 Date: 8/30

Fall 2016

Note Title

Recitations

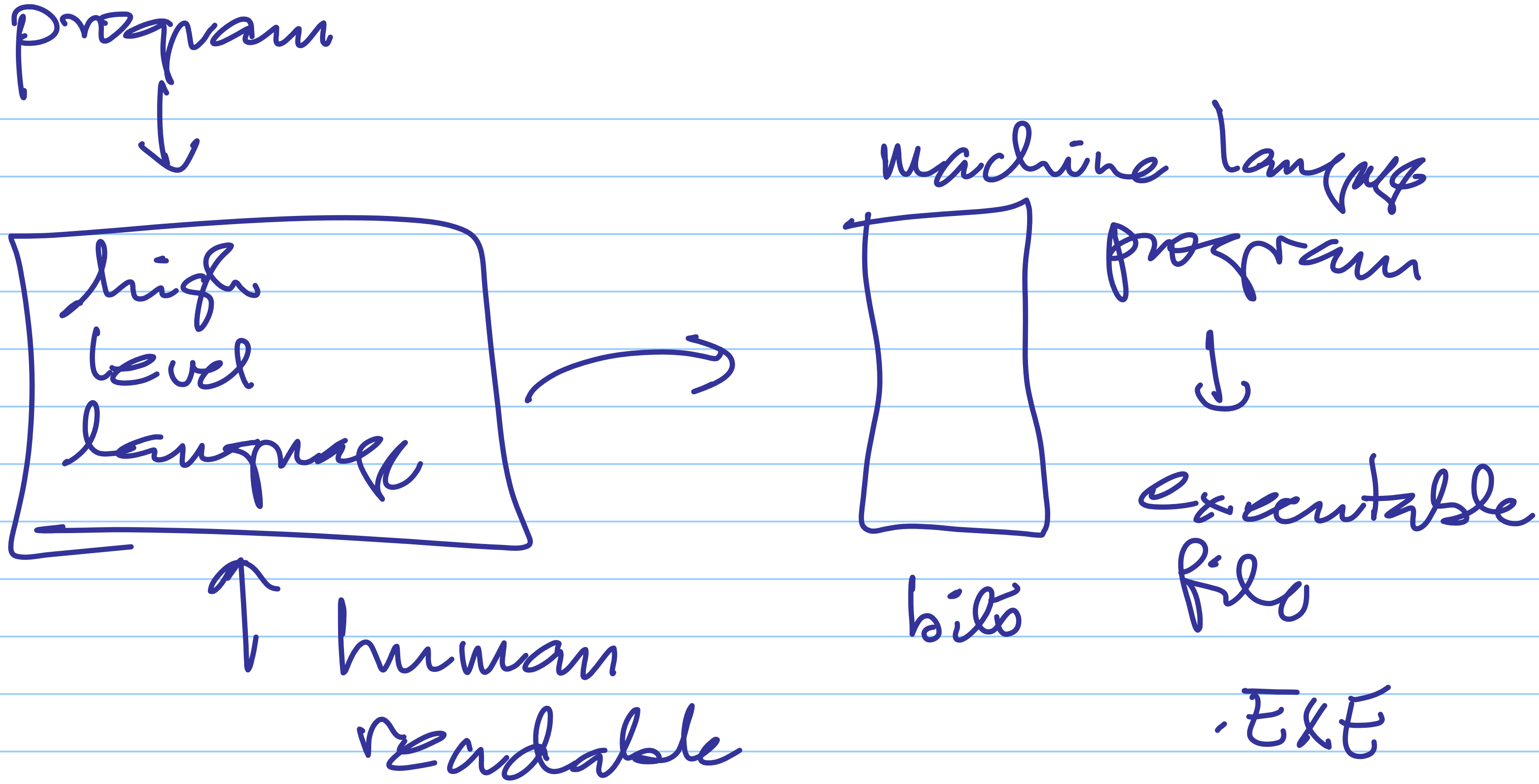
	TUE	12-1p	BYAC 190
new →	WED	12-1p	BYAC 110
	WED	4-5p	BYAC 190

Protection

process - execution

interrupts & system calls

program → execution



program

high level language

human readable

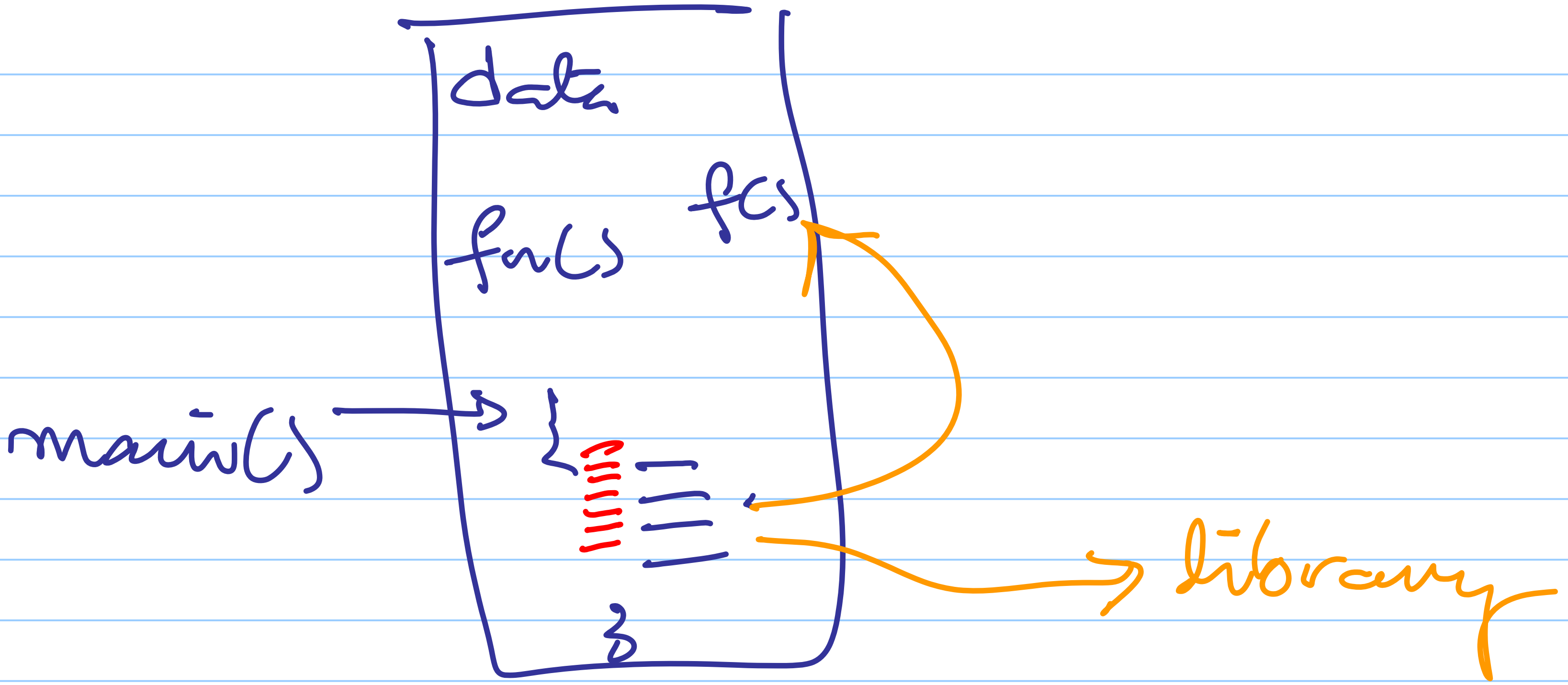
machine language program

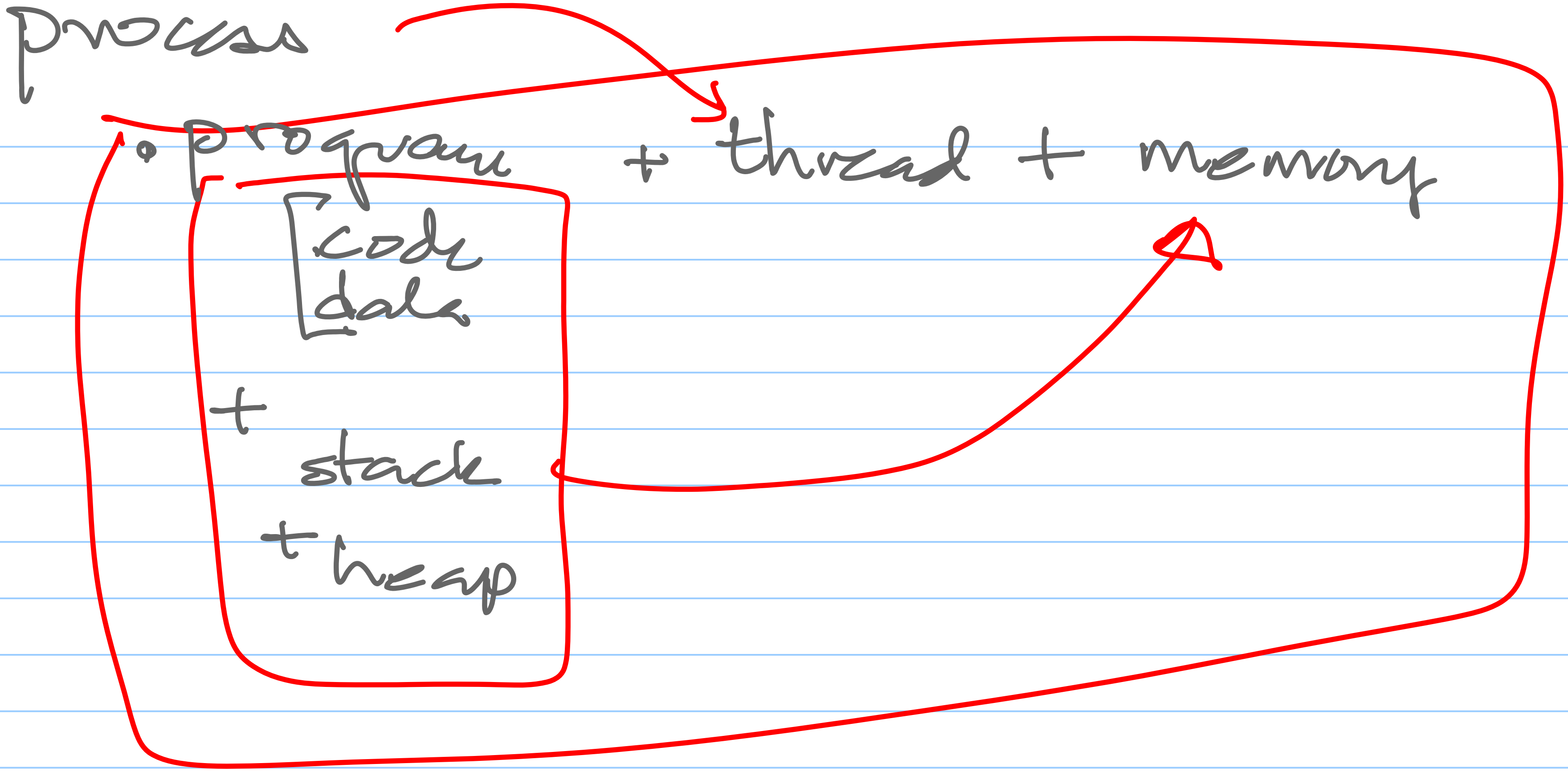
bits

executable file

.EXE

program → process





process

- program + thread + memory

code
data

+ stack

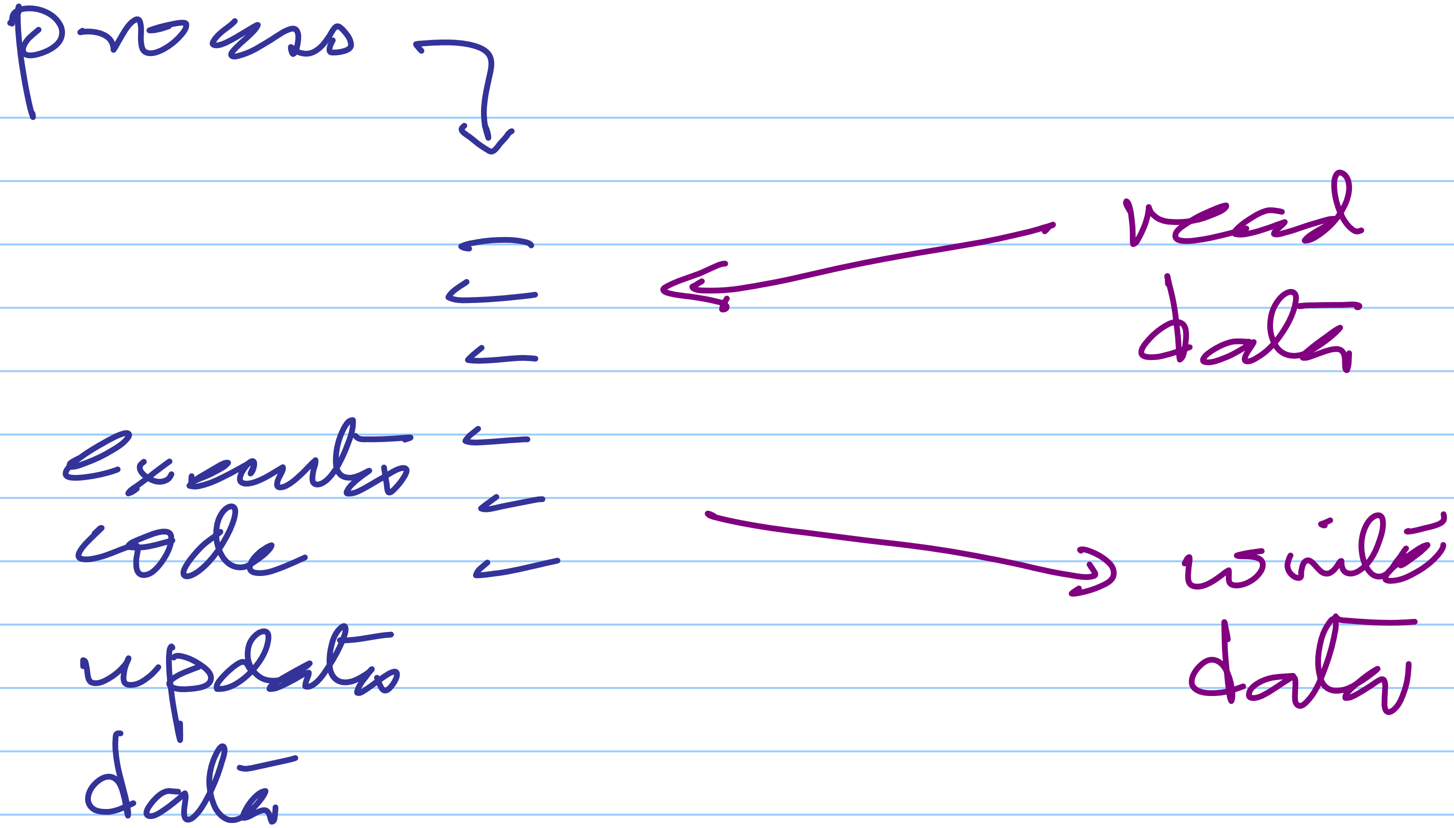
+ heap

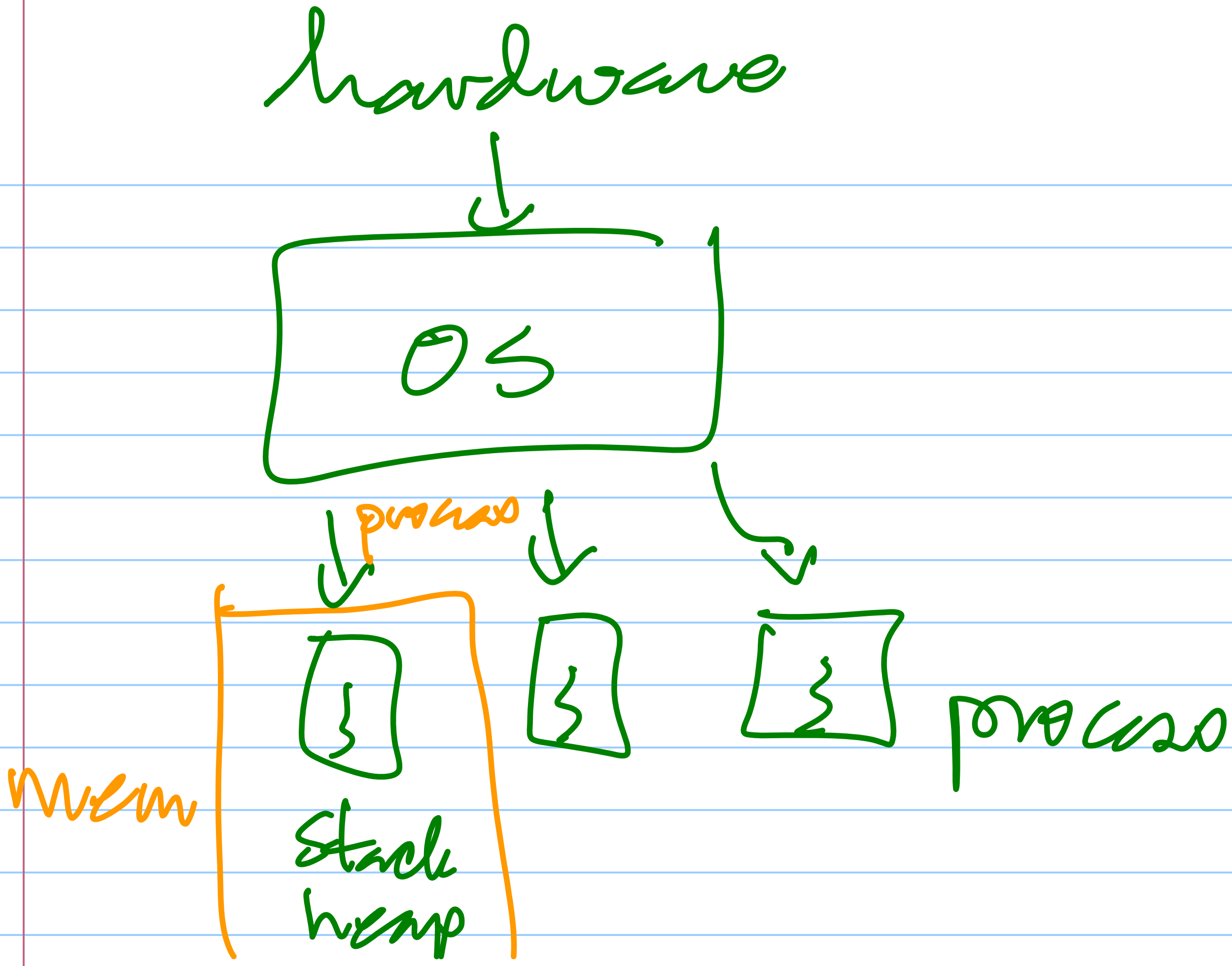
process = prog + stack + heap + thread

program & thread are orthogonal

→ a program may contain more than 0 or more threads

→ a thread may run more than 1 program

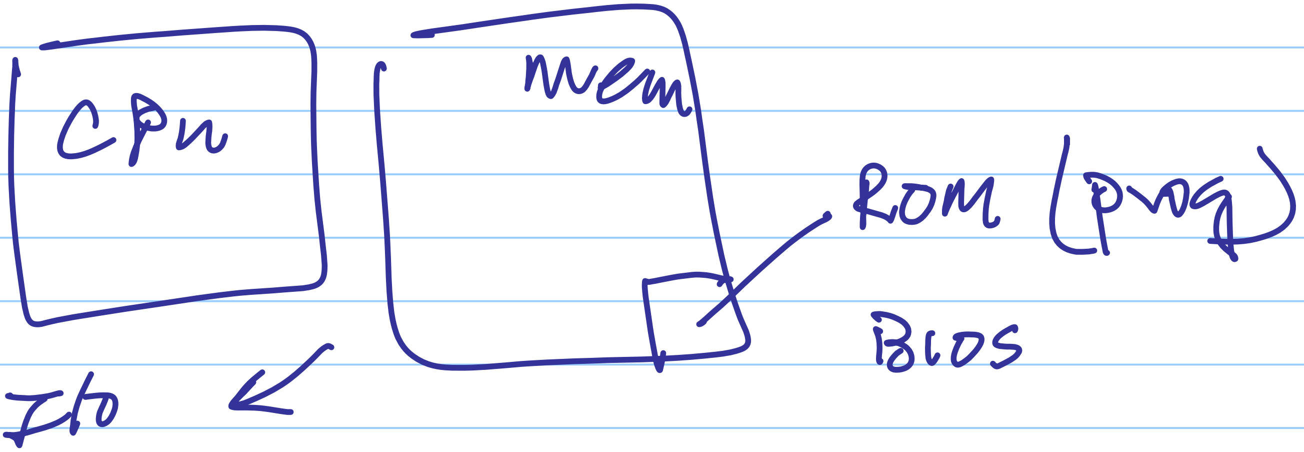

process
read data
execute code update data
write data


hardware
OS
process
process
mem
stack
heap

Process Create processes

1st process

↳ BOOT a computer



CPU → runs a prog in BIOS

→ locates the OS kernel on disk

→ load kernel to RAM

→ jumps to main() of kernel

INIT process

1st process

Init process starts a lot of processes

↳ services

↳ user interface

↳ login process ✓

↳ shell process

etc

use system call

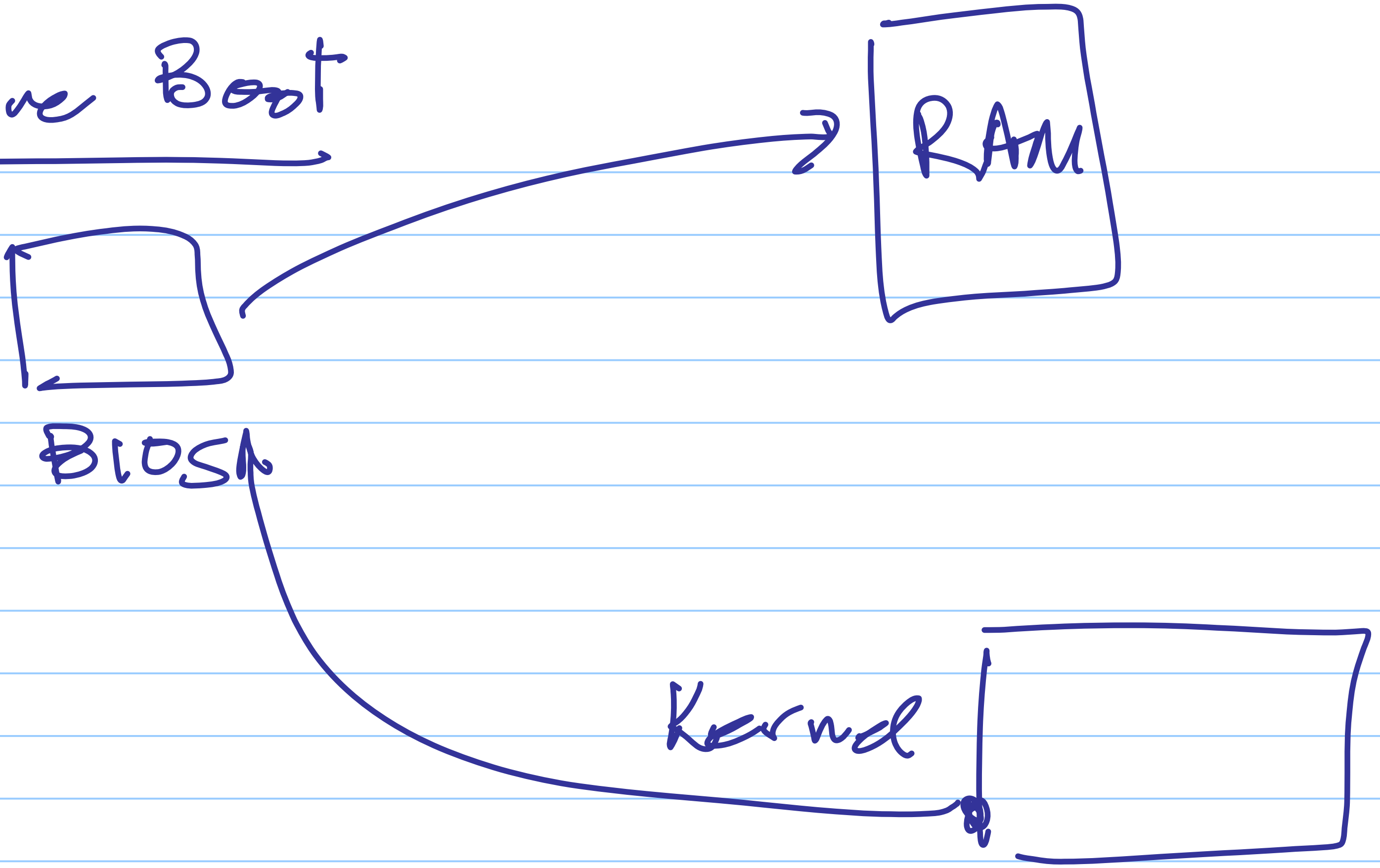
"fork"

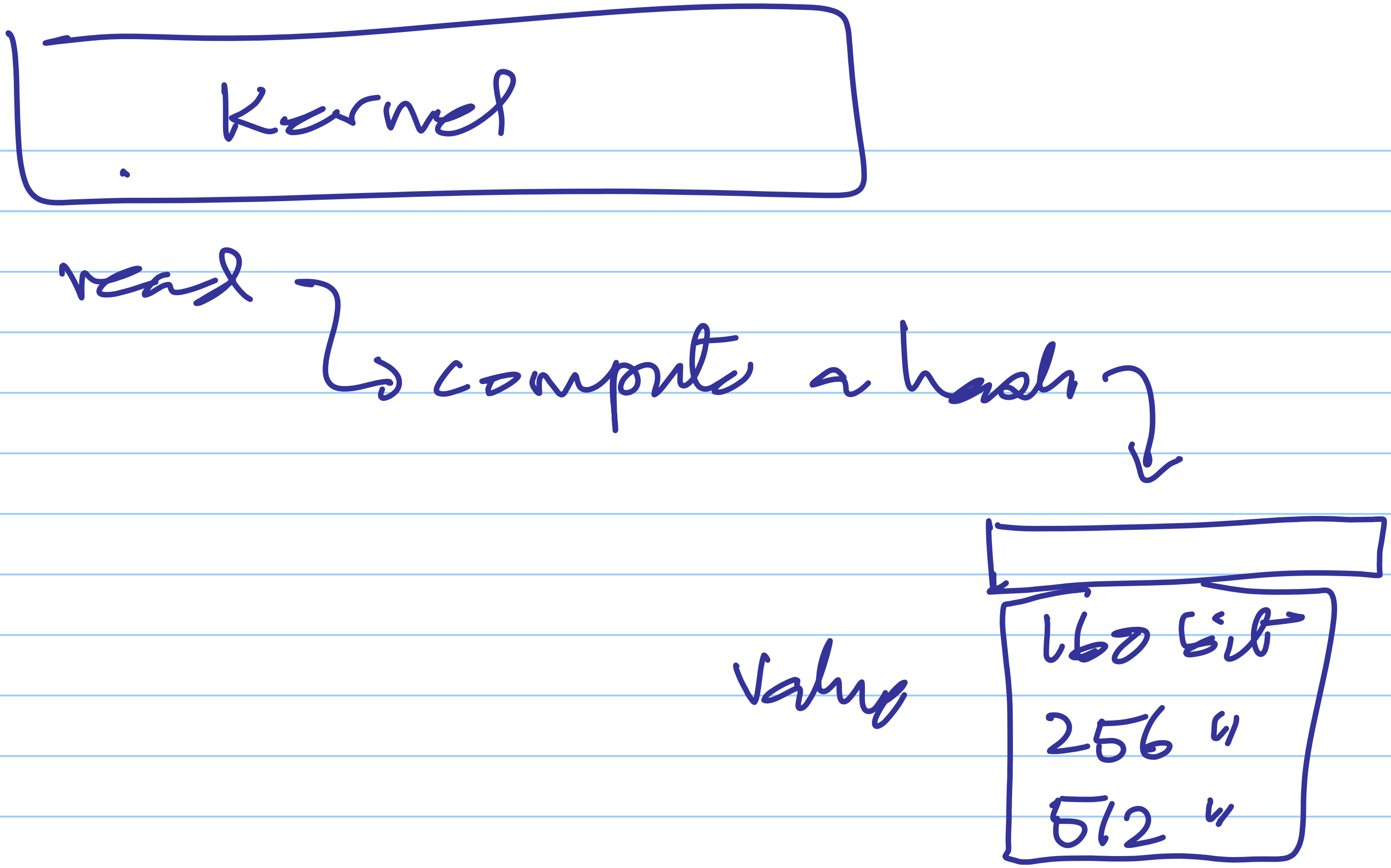
↳ allocate mem

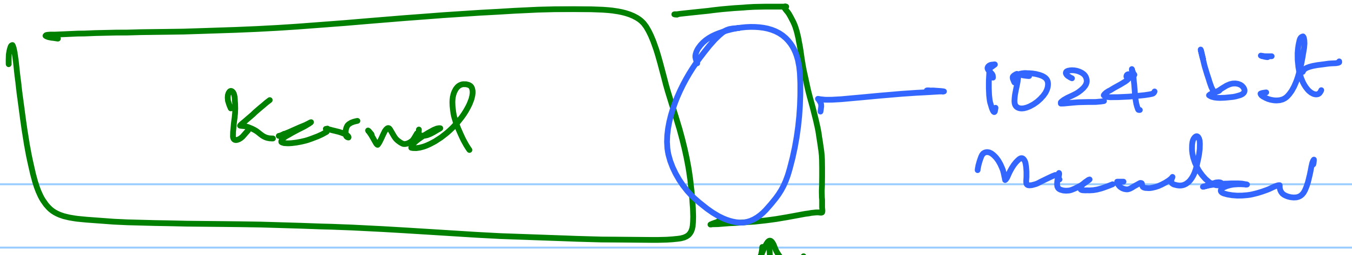
init stack / heap

↳ create thread

Secure Boot




Kernel
read
compute a hash
Value
160 bit
256 "
512 "



hash signed by the original source/corporation

use microsoft public key to verify signature ← does not change + embedded in BIOS protection → mem + CPU modes

- Secure boot

- user process privilege levels →

  ↳ user → u1, u2, u3 ...
  privileged. (root)

→ syscall → OS checks allowable

Vulnerabities, malware

↳ how a malware gets installed

↳ social engineer

↳ buffer overflow

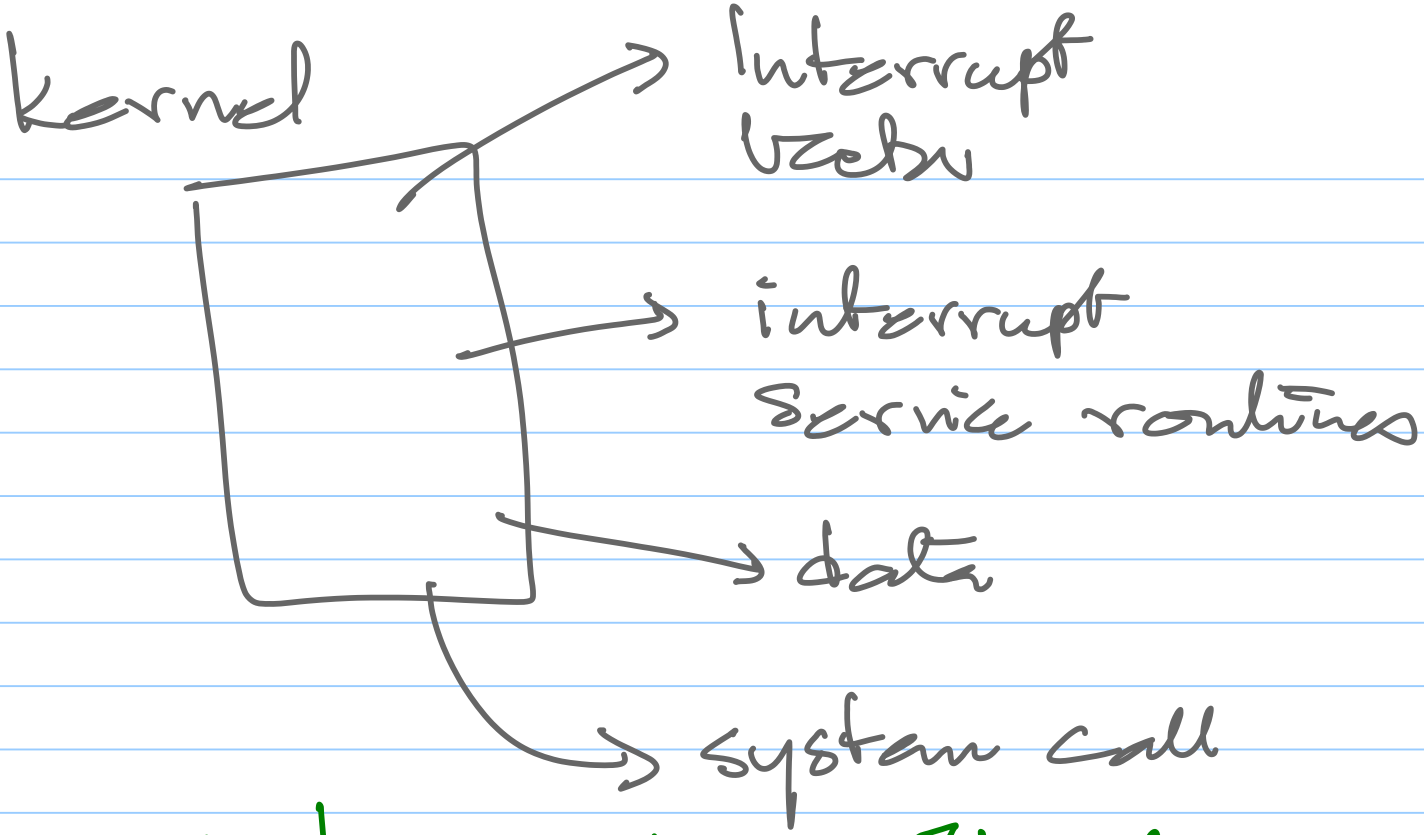
↳ programs that are malicious

⇒ new buffer overflow discovery

↳ zero day exploit

↳ reveal it to the vendor

→ exploit → patch.



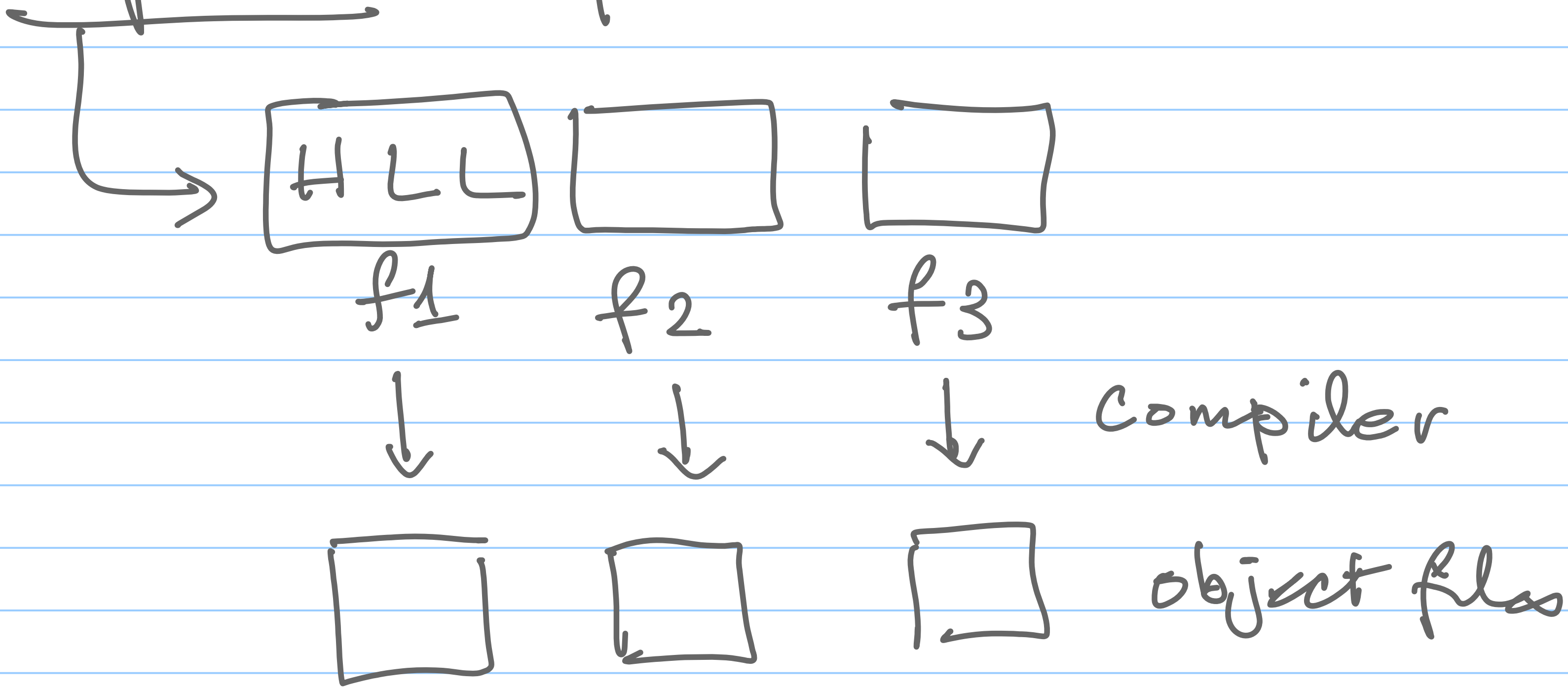
+ device drivers [hardware specific]

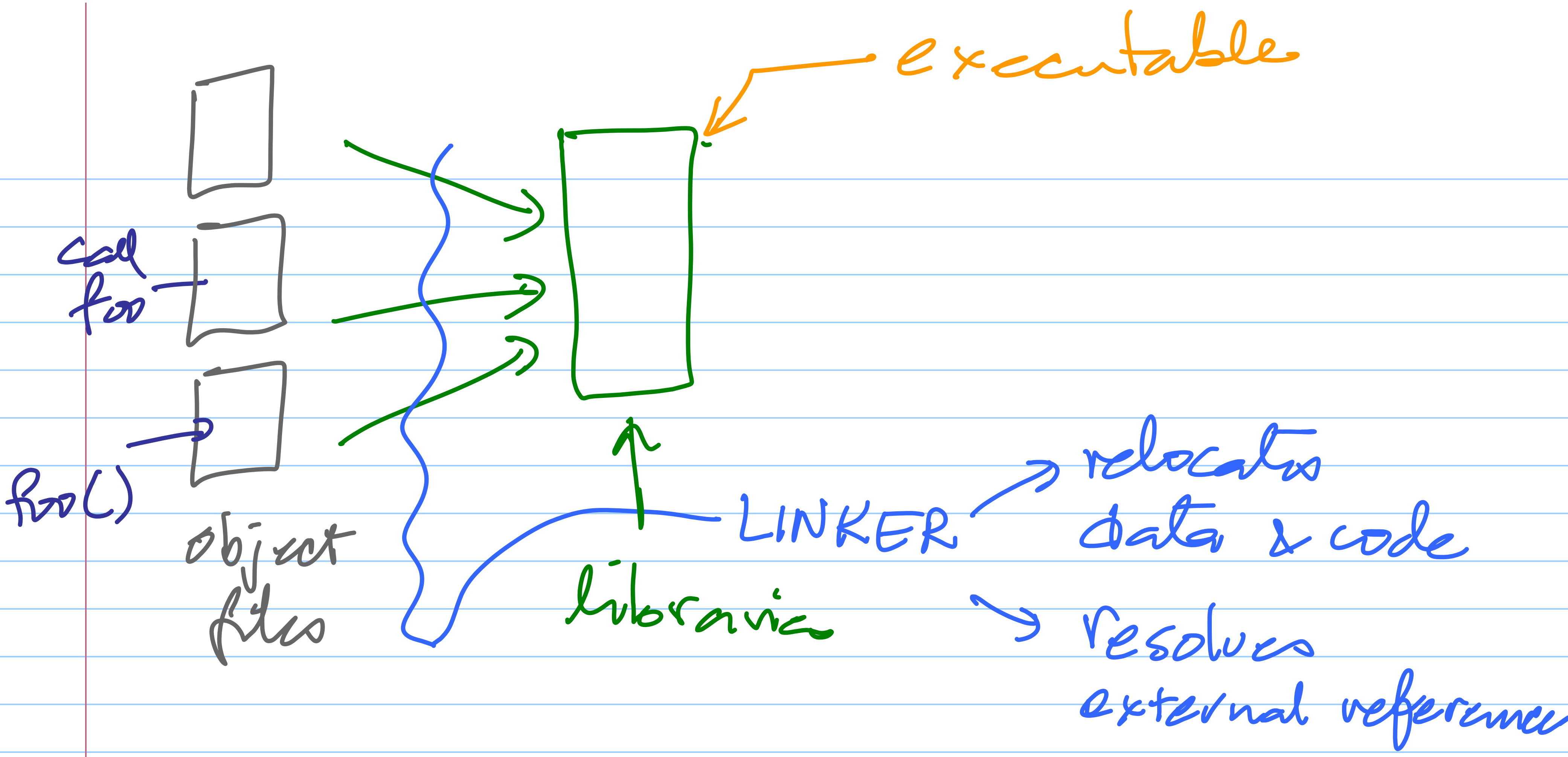
kernel can be configured with new drivers

↳ load a driver (vulnerable)

→ signed drivers

programs & processes

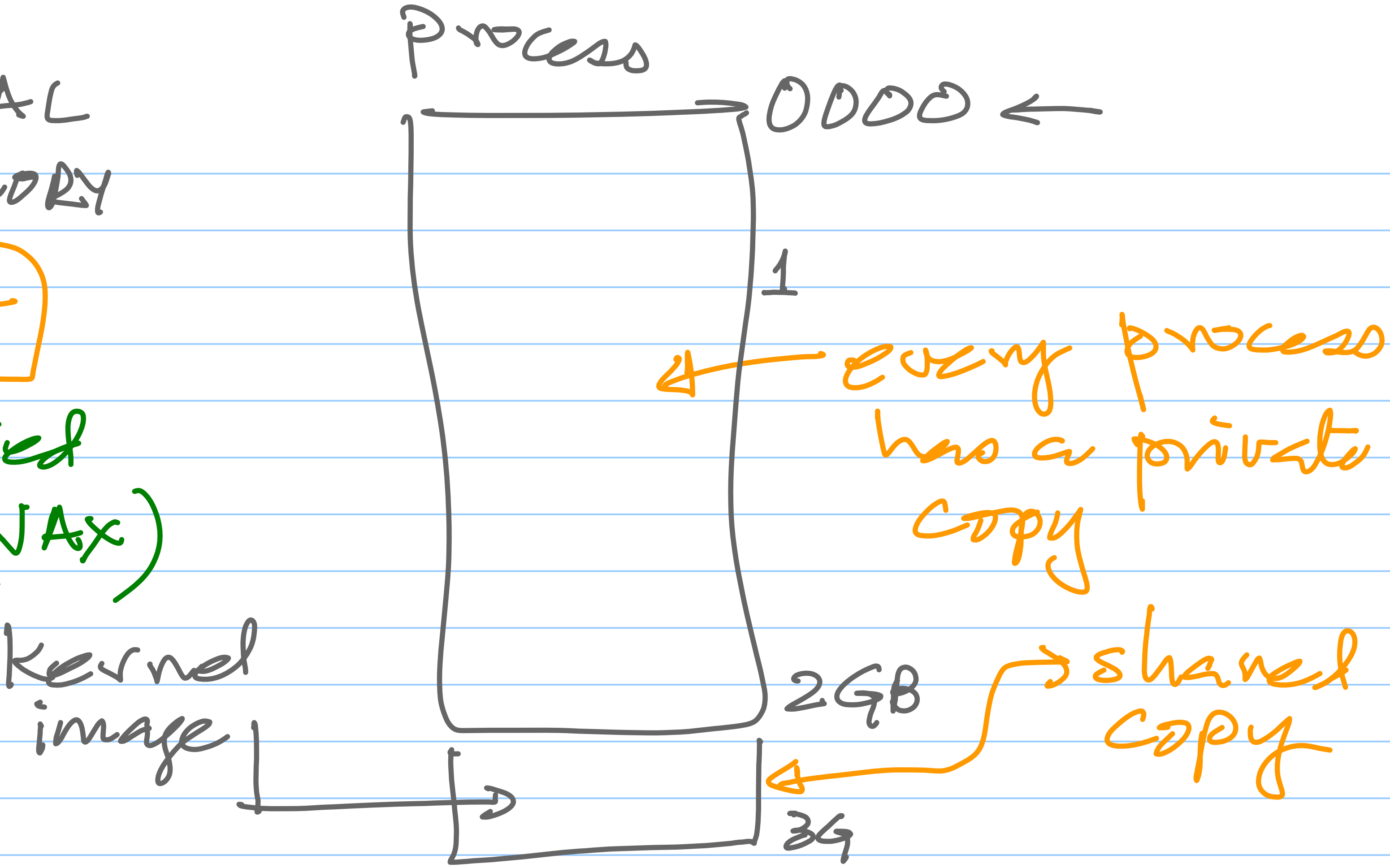



executable
call foo
foo()
object files
libraries
LINKER
relocates data & code
resolves external references

VIRTUAL MEMORY

32 bit

modified from (VAX)


Process
0000
1
every process has a private copy
2GB
shared copy
3G
Kernel image