NOTETAKERS NEEDED

VOLUNTEER TO BE A NOTETAKER

We are looking for a volunteer notetaker for this class. If you would like to volunteer please email notetaker@uwindsor.ca


Ezekiel 25:17, Pulp Fiction (1994), Quentin Tarantino
You read the Bible, Brett?

LEC02

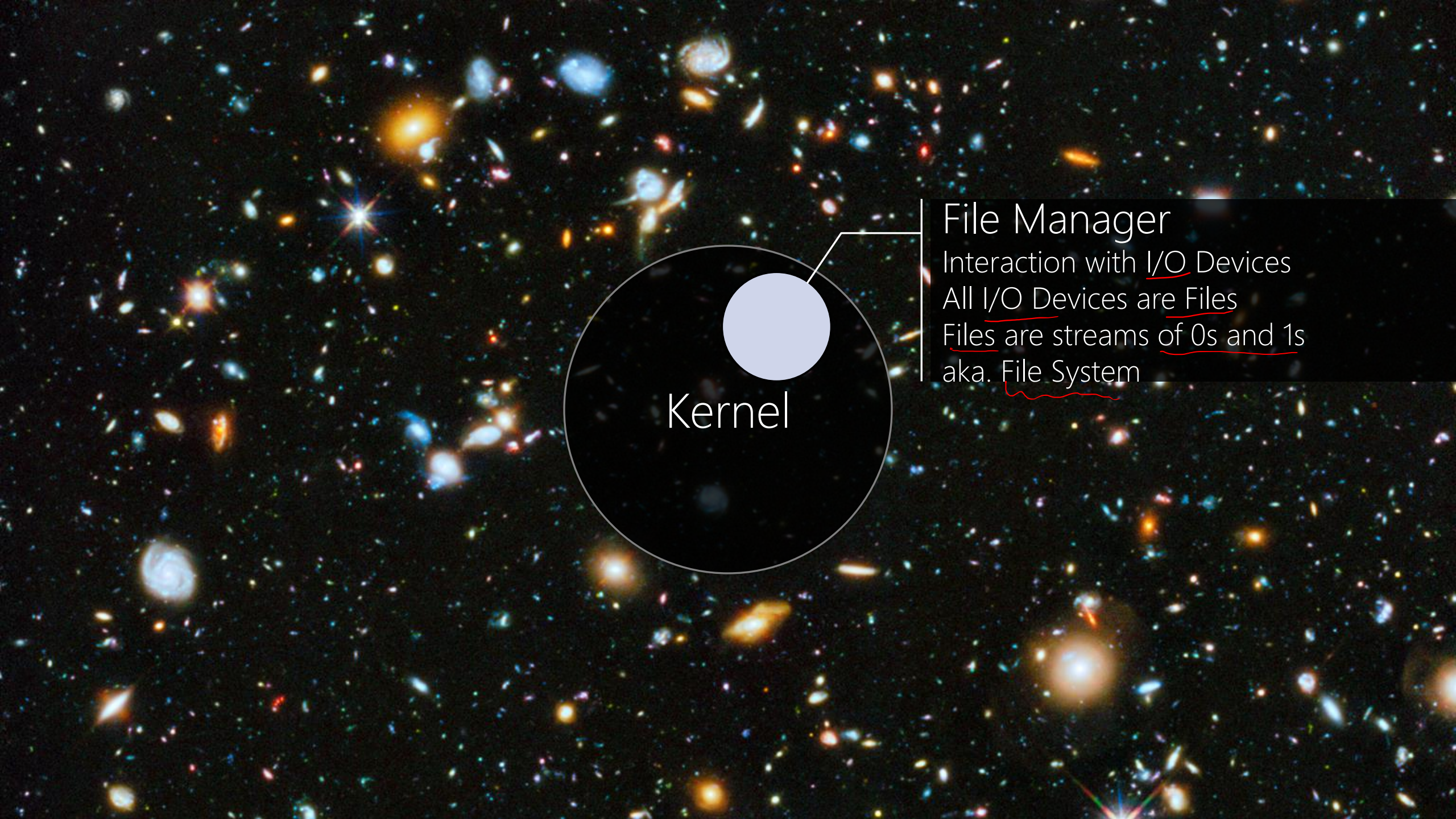
W#

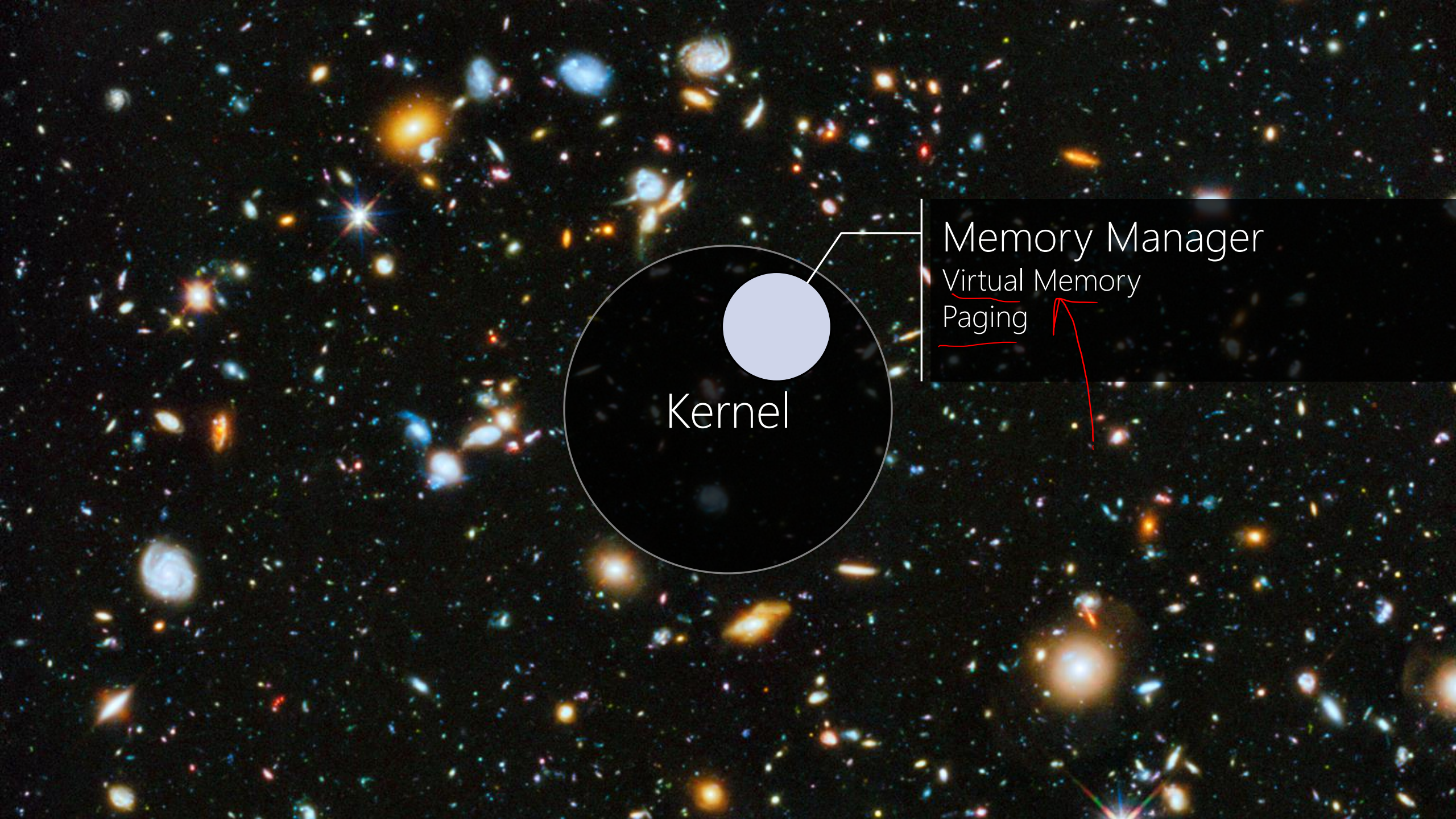
Lecs >> Lec02: Intro to System Programming

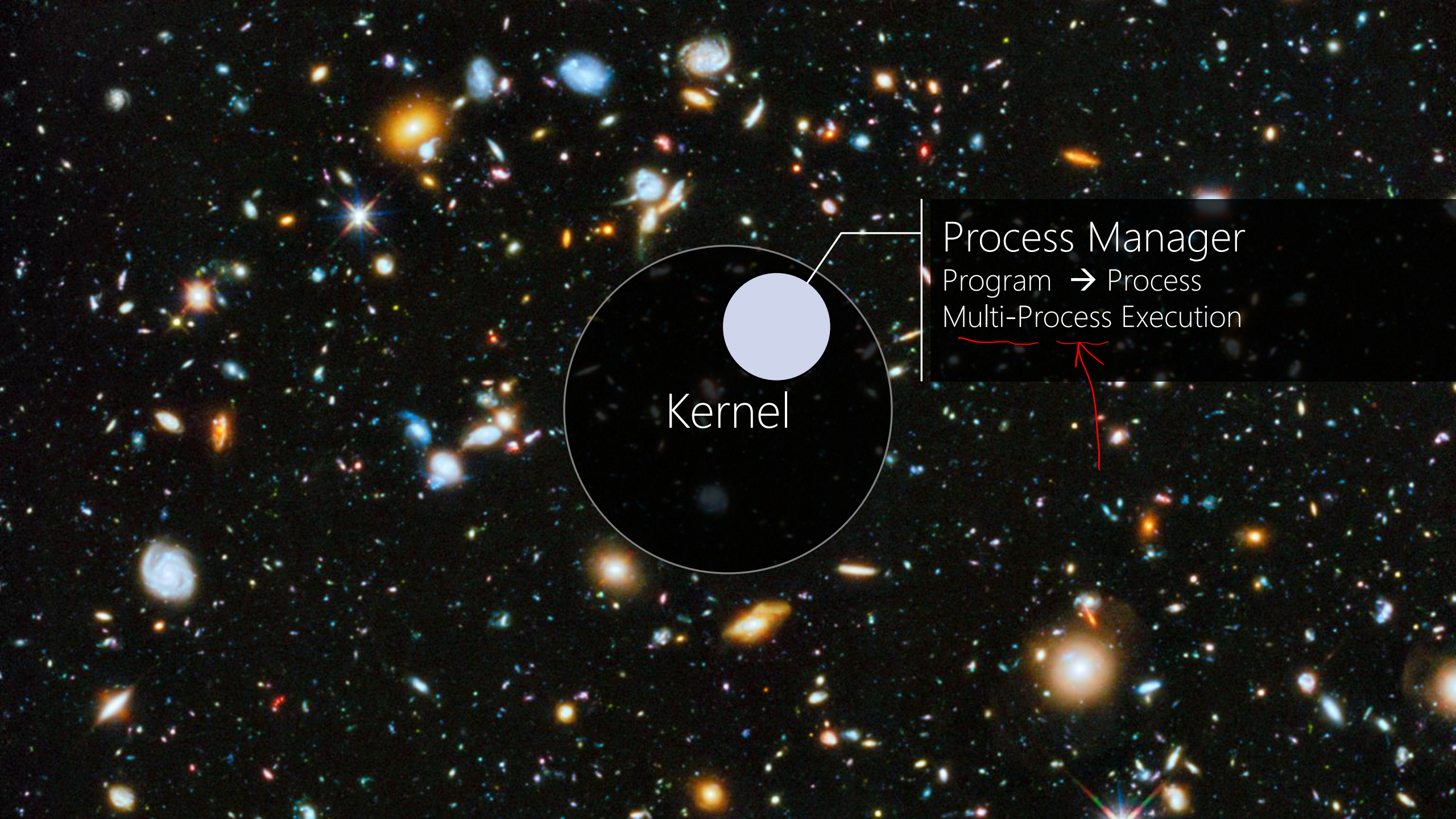
LAB02

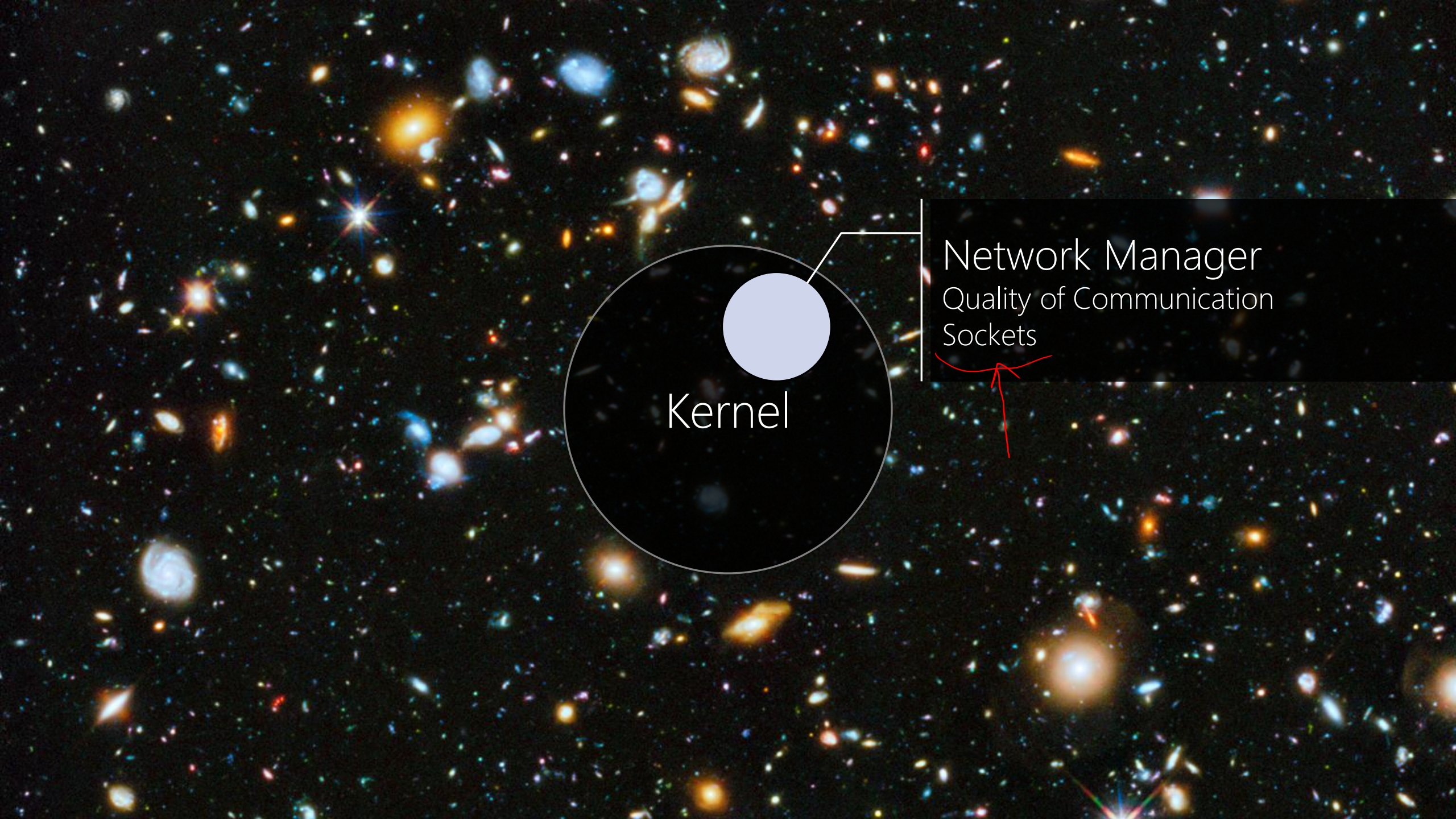
Labs >> Lab02: Working Environment Setup

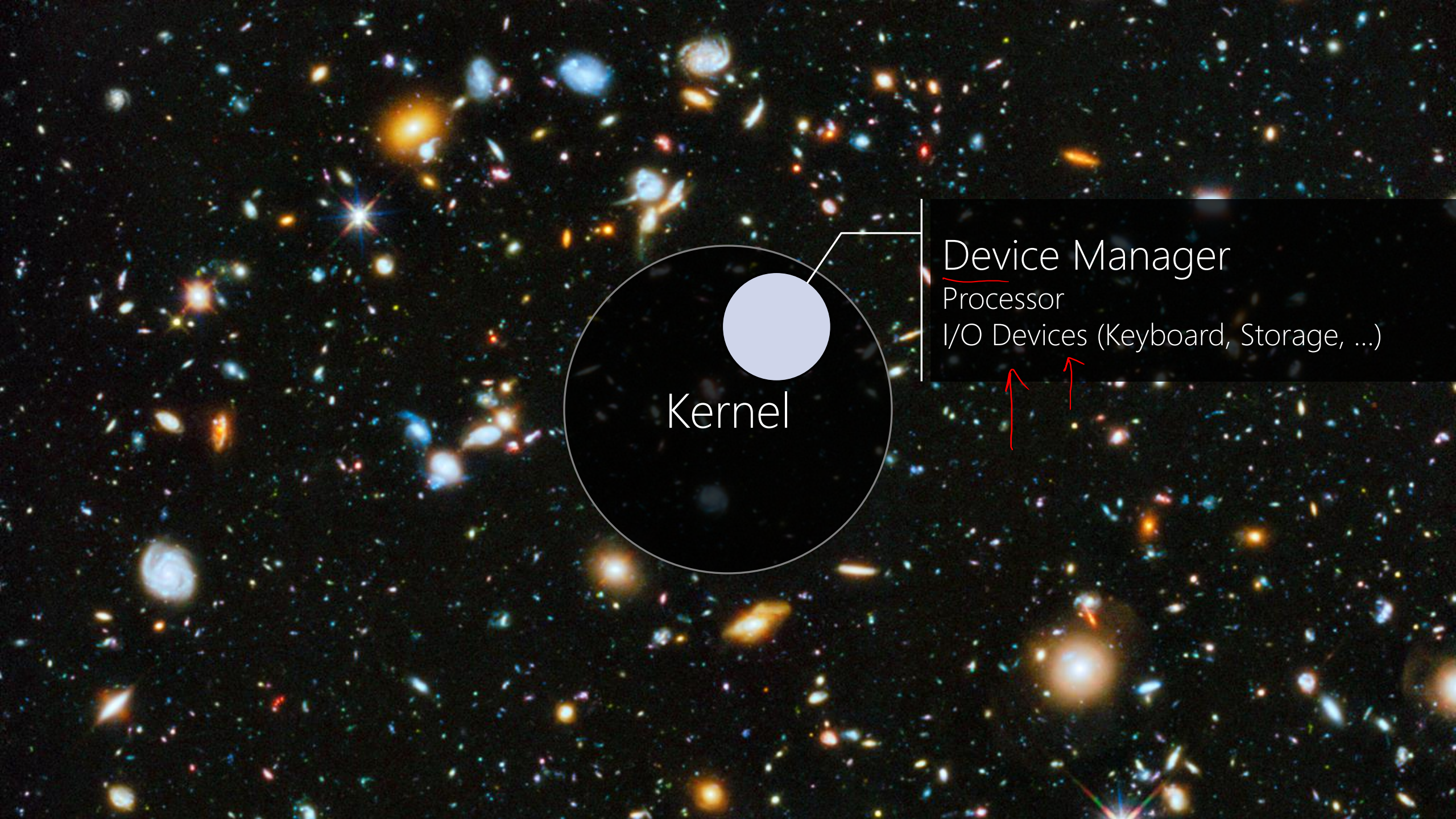

UNIX
Kernel

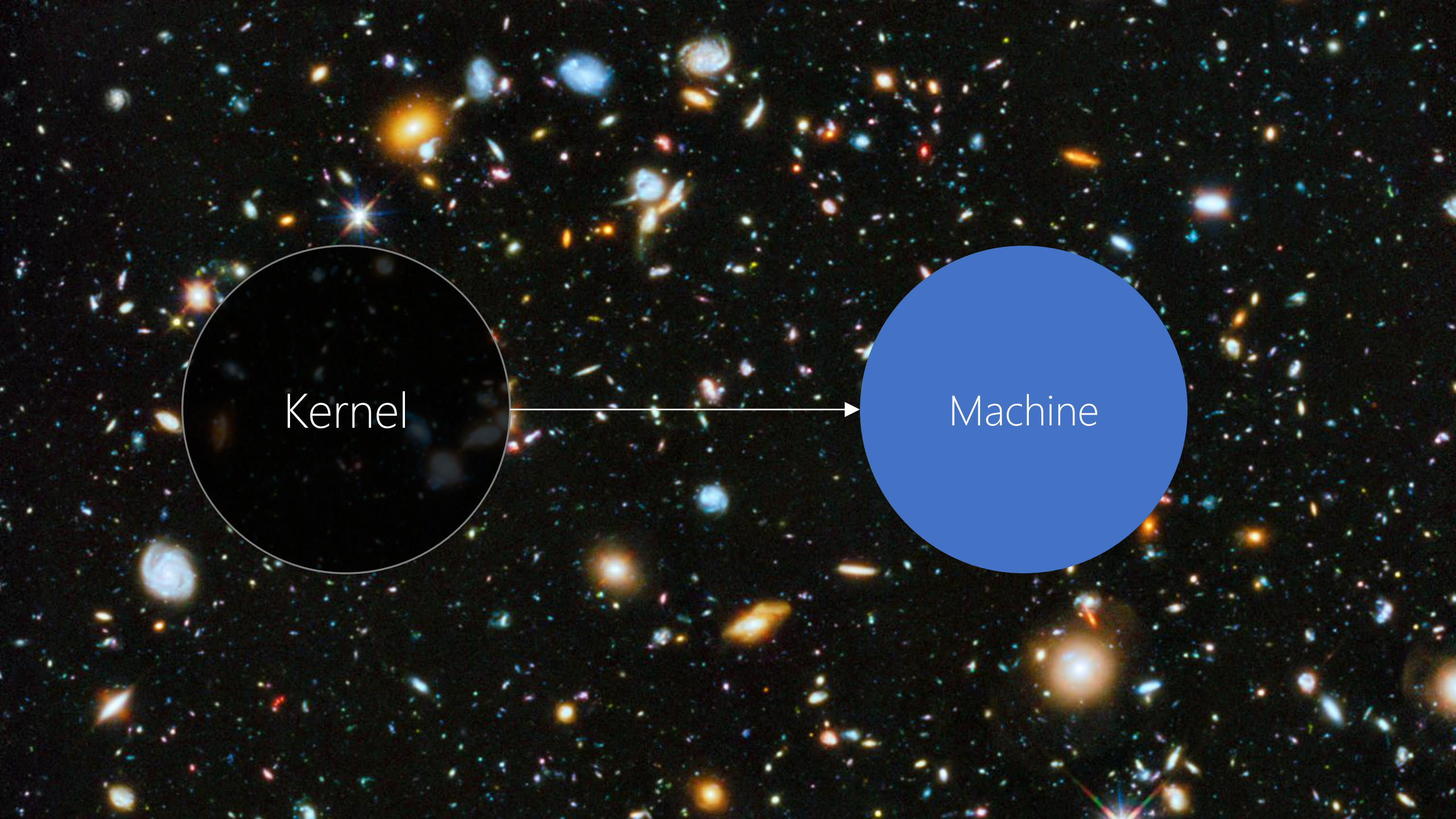

Kernel
File Manager
Interaction with I/O Devices
All I/O Devices are Files
Files are streams of 0s and 1s
aka. File System


Kernel
Memory Manager
Virtual Memory
Paging


Kernel
Process Manager
Program → Process
Multi-Process Execution


Kernel
Network Manager
Quality of Communication
Sockets


Kernel
Device Manager
Processor
I/O Devices (Keyboard, Storage, ...)

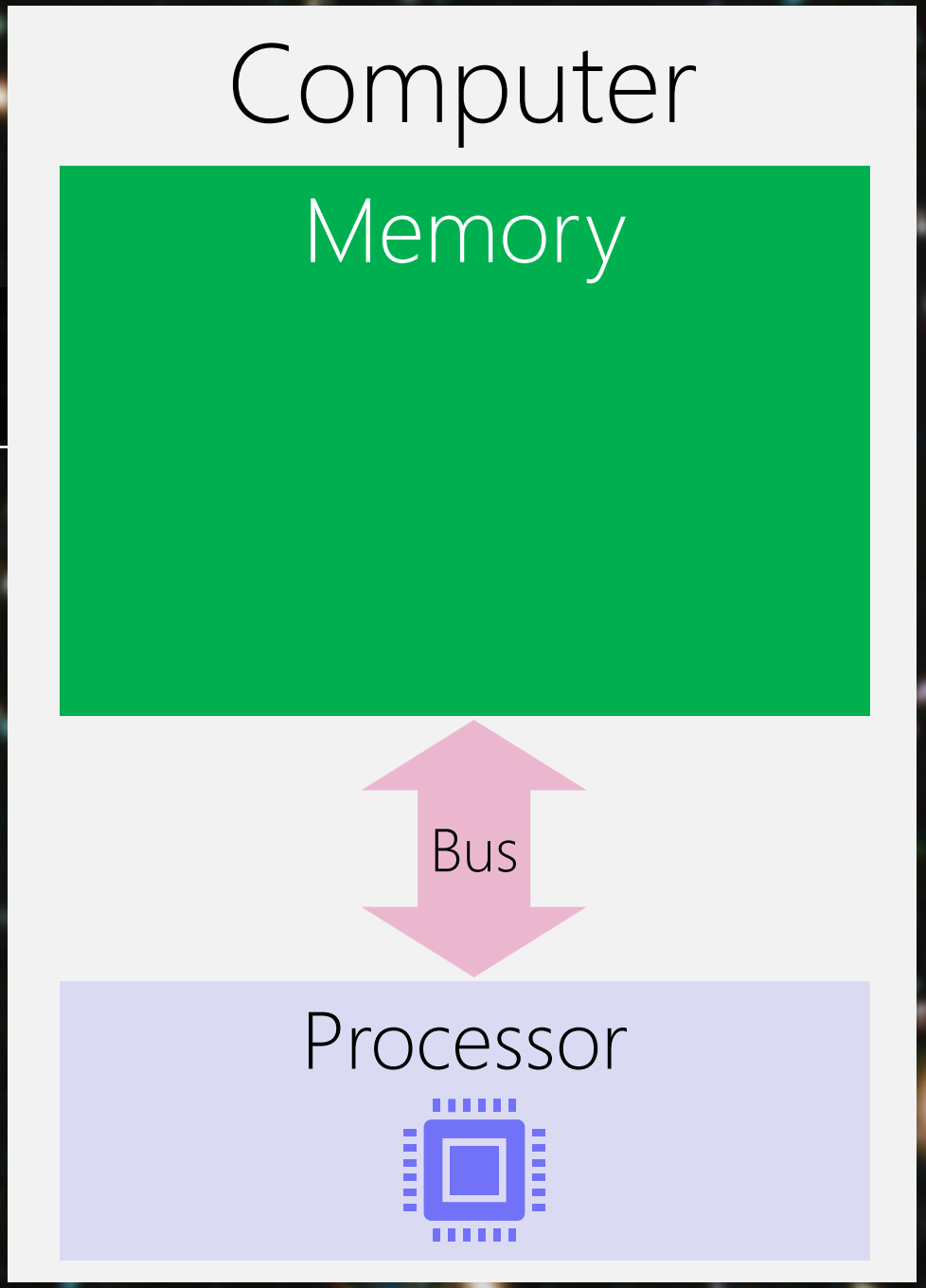

Kernel
Machine

Kernel

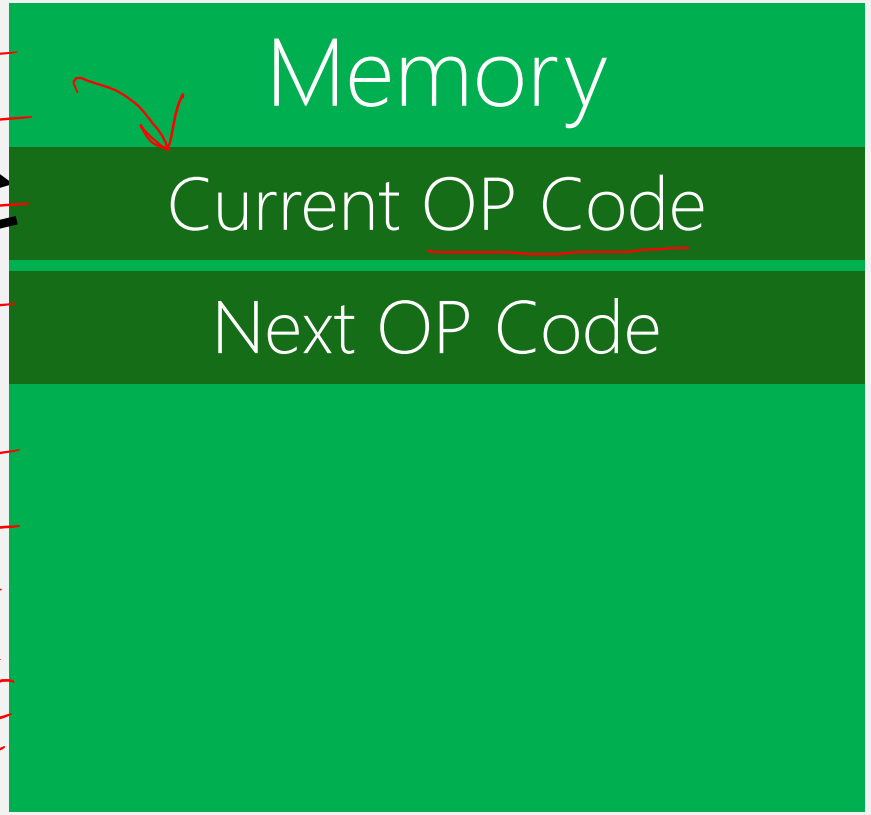
Bootup

Boot is short for bootstrap

pull oneself up by one's bootstraps



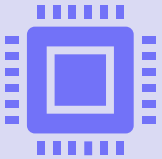
Computer



1. Fetch

Must be an instruction
Otherwise, raise error

Processor

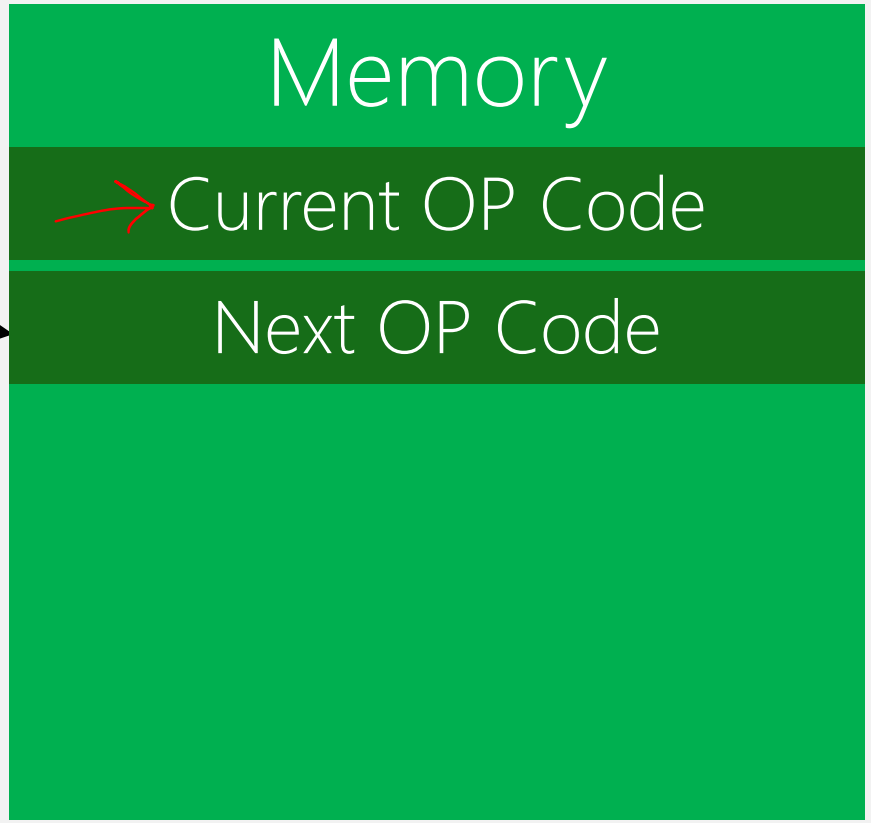


Instruction Pointer (IP)
aka Program Counter (PC)

10110

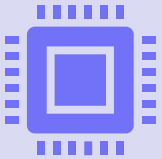
Computer

2. IP = IP + 1



Instruction Pointer (IP)
aka Program Counter (PC)

Processor

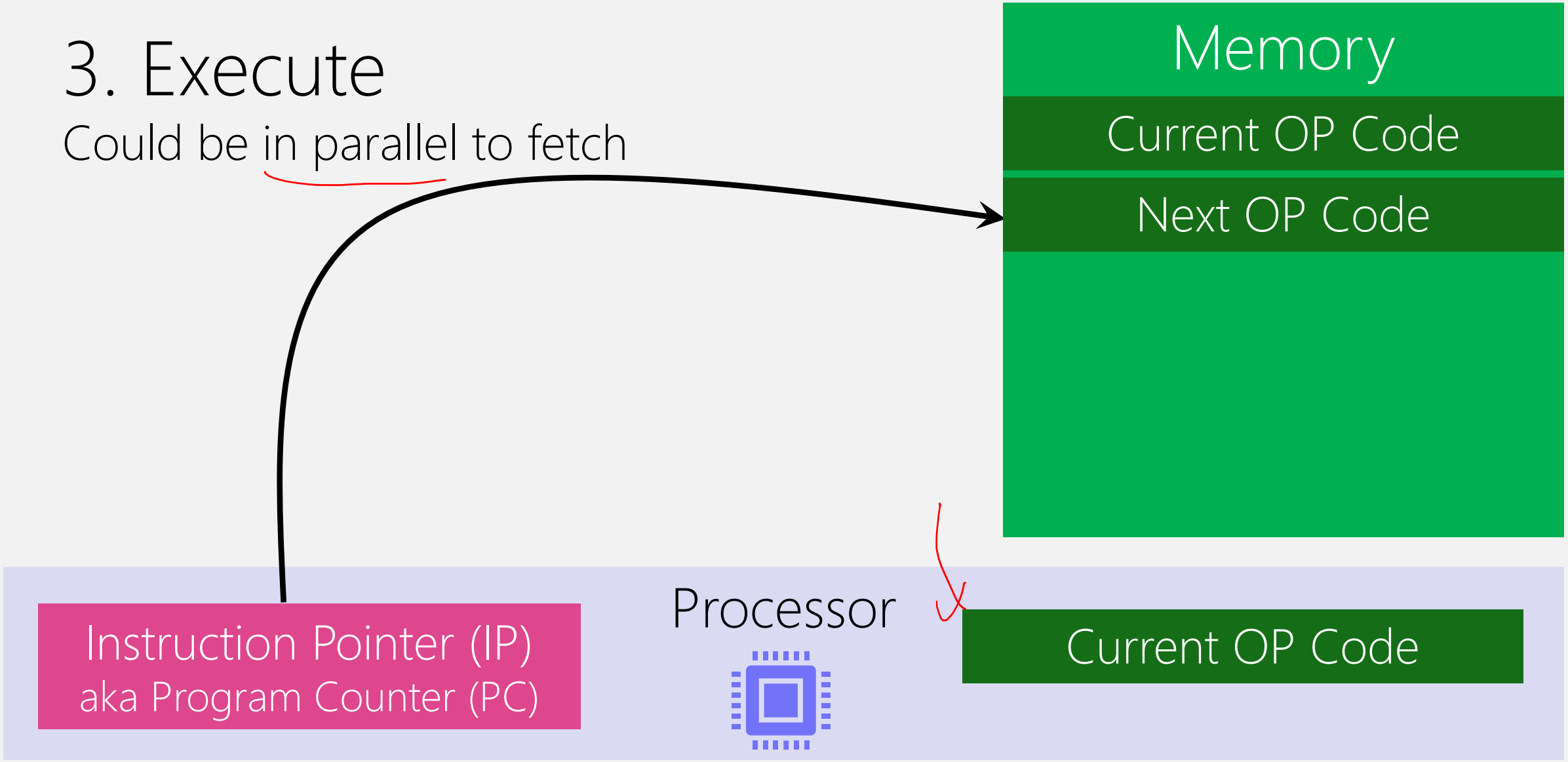


Current OP Code

Computer

3. Execute

Could be in parallel to fetch



Computer

1. Fetch

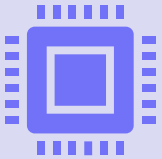
Memory

Current OP Code

Next OP Code

Processor

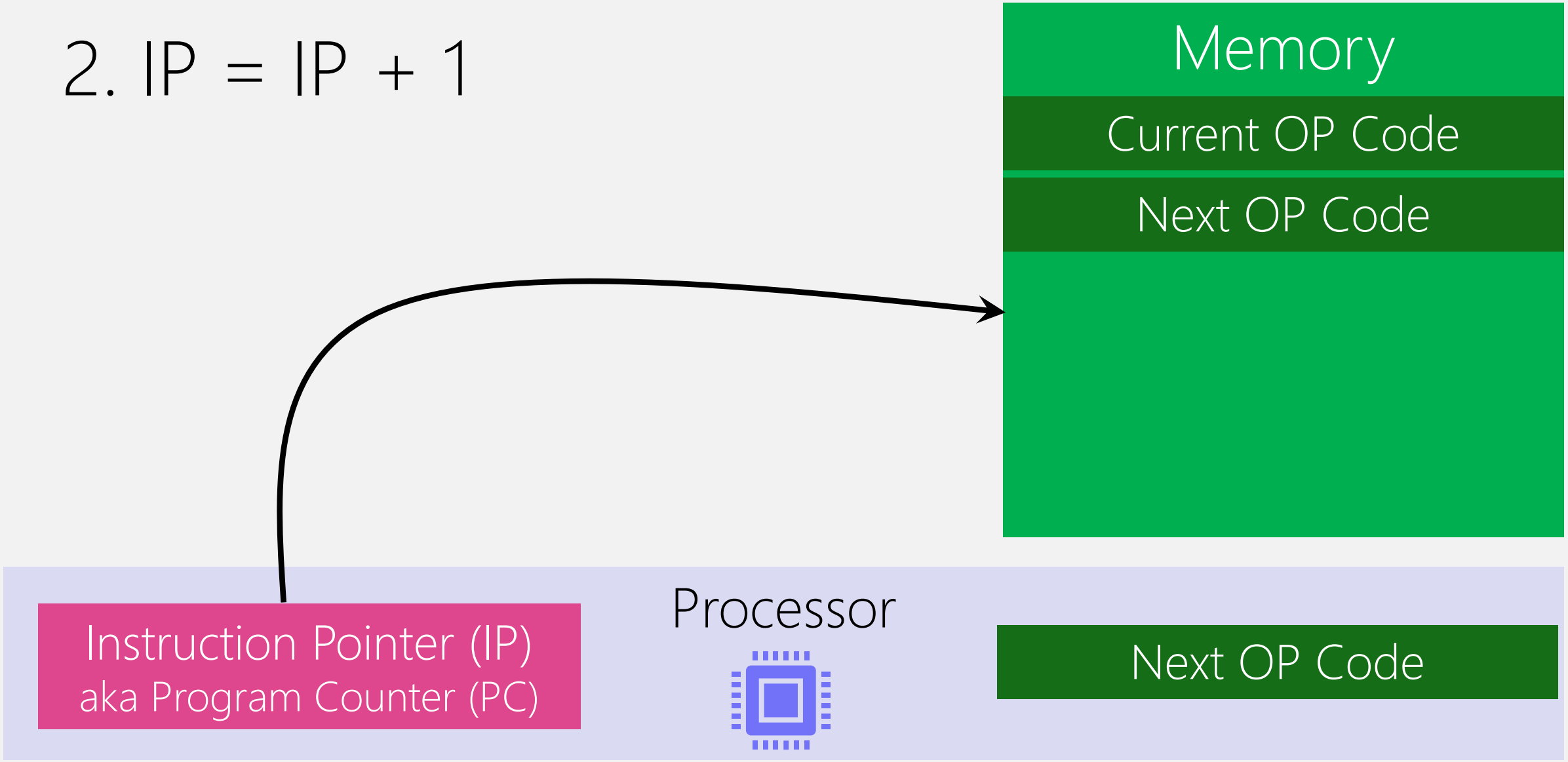
Instruction Pointer (IP)
aka Program Counter (PC)



Next OP Code

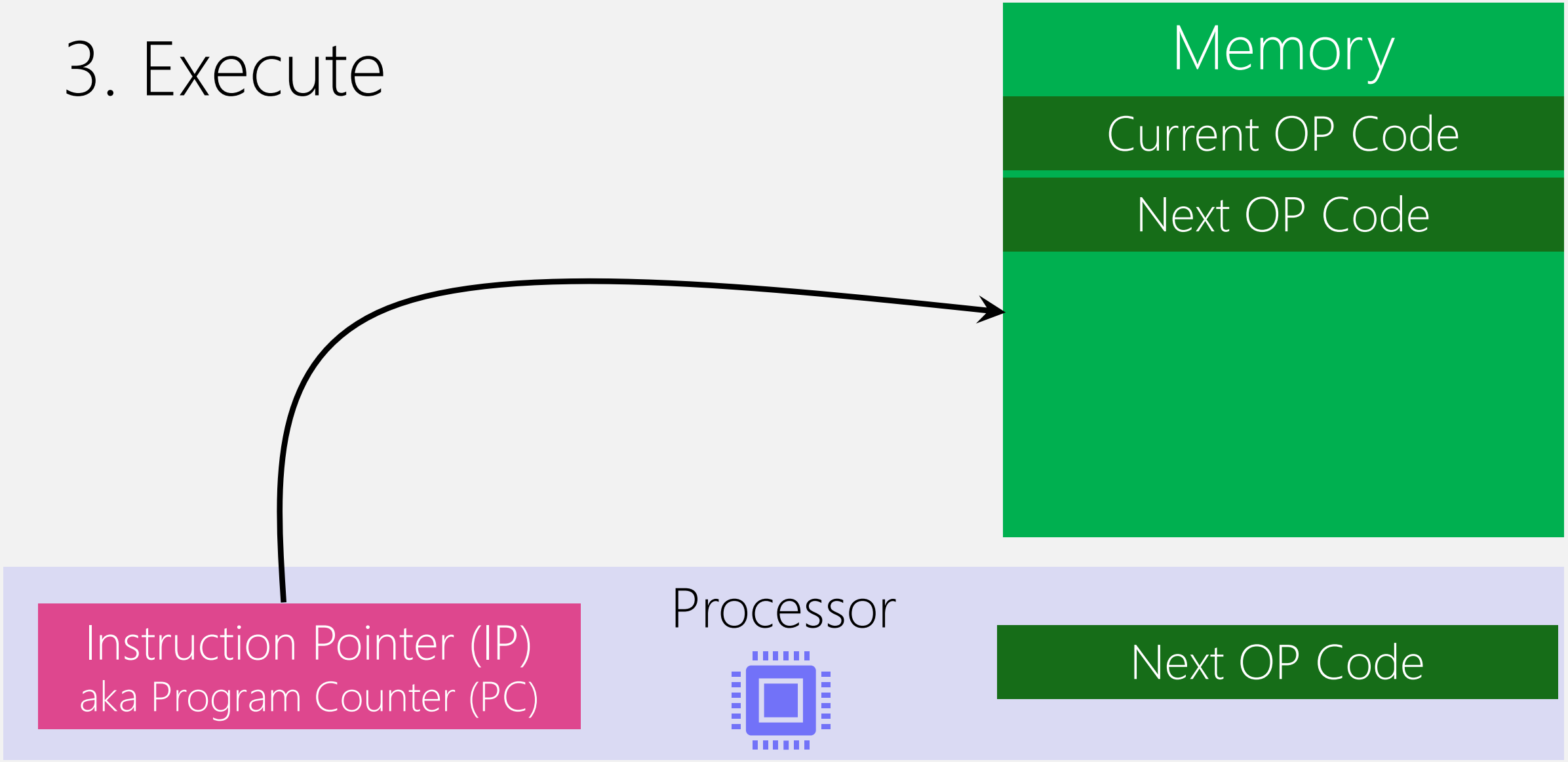
Computer

2. IP = IP + 1

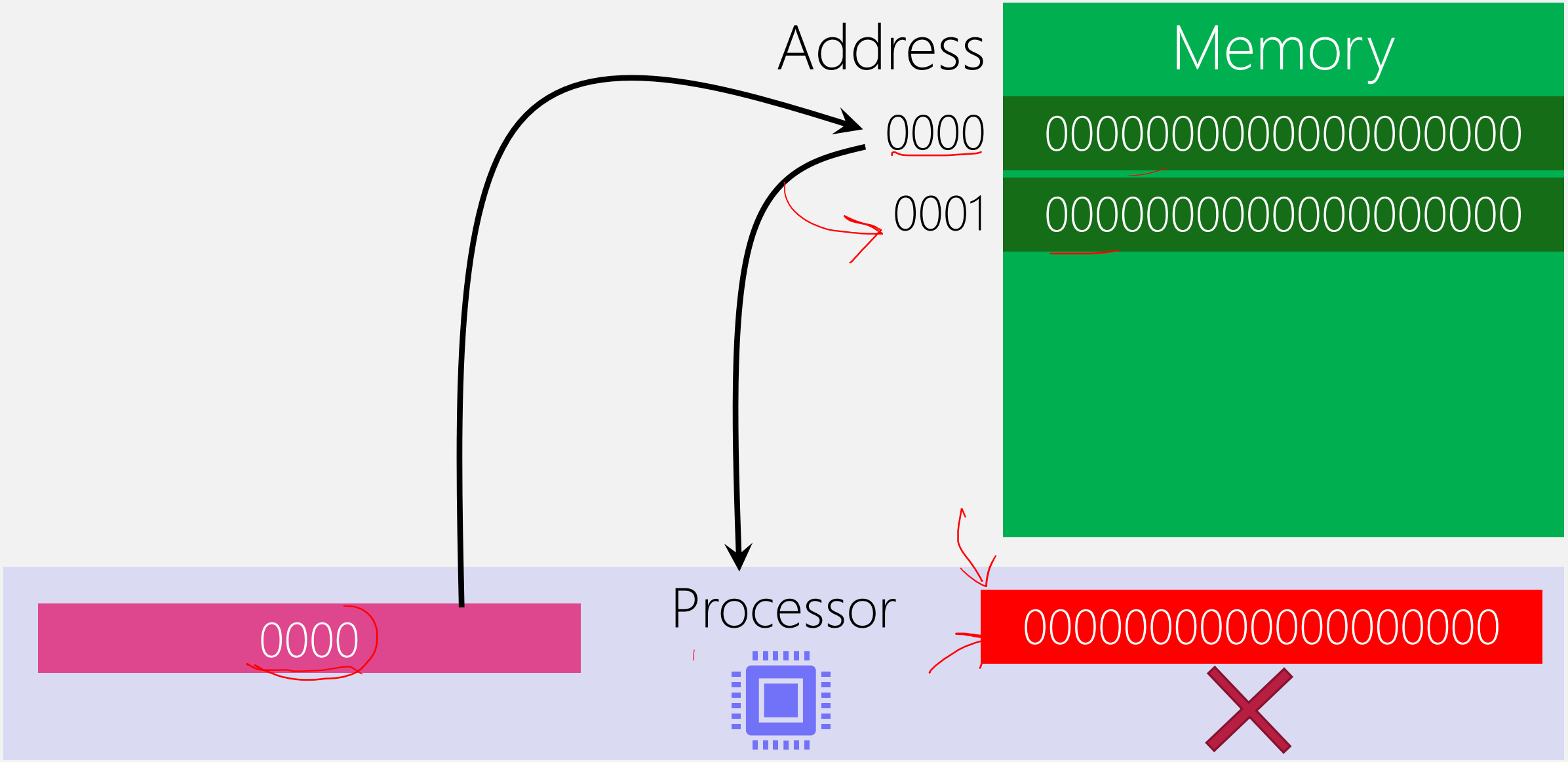


Computer

3. Execute



Turn ON


Address
Memory
0000
0000000000000000000
0001
0000000000000000000
0000
Processor
0000000000000000000

What happens?

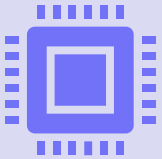
IP = 0000 → mov IP, 000

Address	Memory
0000	IP = 0000
0001	Next OP Code

0001 → 0000

0000

Processor

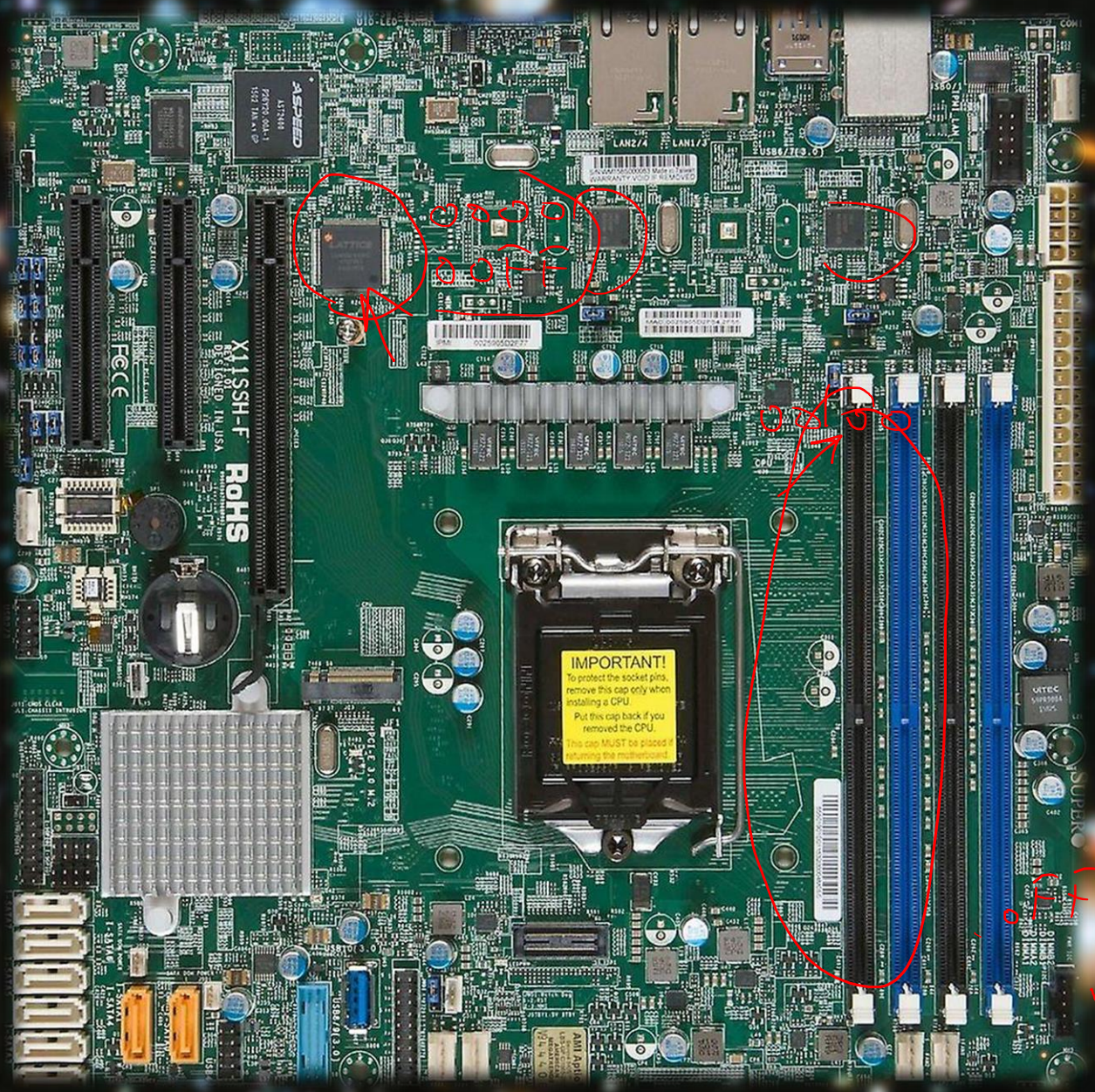


IP = 000


Busy-Waiting
Busy-Looping
Spinning
Loveless (2017), Andrey Zvyagintsev

BIOS

Basic I/O System

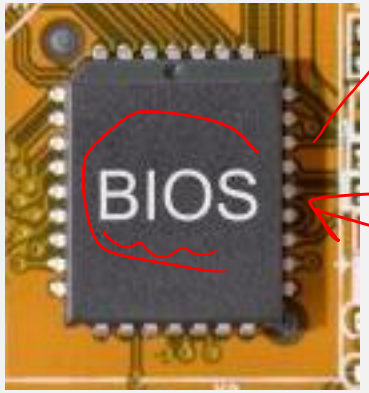

X11SSH-F
RoHS
IMPORTANT!
To protect the socket pins, remove this cap only when installing a CPU.
Put this cap back if you removed the CPU.
This cap MUST be placed if returning the motherboard.
LAN2/4
LAN1/3
ASPEED
SUPERO

RoM

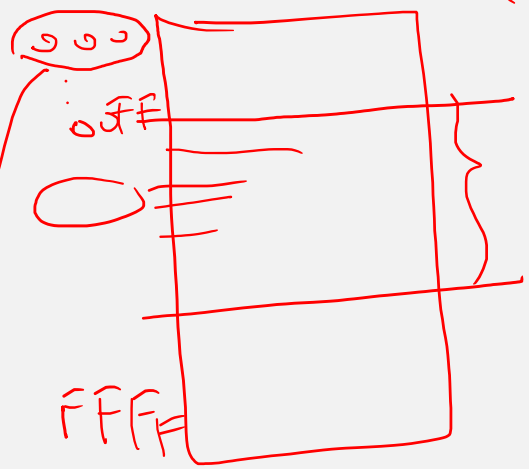
OFFF


F12 Boot Options
DELL

Basic I/O System (BIOS)

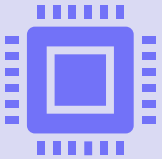


Address	Memory
0000	BIOS: Check
0001	BIOS: Load MBR
0010	BIOS: IP = &MBR



Processor

0000



Basic I/O System (BIOS)

Address	Memory
0000	BIOS: Check
0001	BIOS: Load MBR
0010	BIOS: IP = &MBR

0000 ~~0001~~ FF00 FF01

Processor

IP = FF00

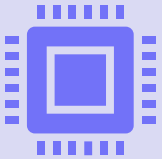
~~BIOS: Check~~

Basic I/O System (BIOS)

Address	Memory
0000	BIOS: Check
0001	BIOS: Load MBR
0010	BIOS: IP = &MBR

0001

Processor

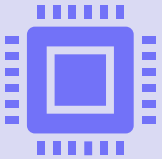


BIOS: Load MBR

Basic I/O System (BIOS)

Address	Memory
0000	BIOS: Check
0001	BIOS: Load MBR
0010	BIOS: IP = &MBR
0100xxx0	MBR: Load Kernel
xxx1	MBR: IP = &Kernel

Processor



0010

BIOS: IP = &MBR

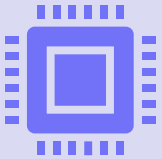
An Instance of Jump!

IP = *&destination*

Address	Memory
0000	BIOS: Check
0001	BIOS: Load MBR
0010	BIOS: IP = &MBR
xxx0	MBR: Load Kernel
xxx1	MBR: IP = &Kernel

0010

Processor



BIOS: IP = &MBR

MBR

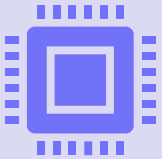
Master Boot Record
Knows where to find the operating system
HDD, DVD, USB, …

Master Boot Record (MBR)

Address	Memory
0000	BIOS: Check
0001	BIOS: Load MBR
0010	BIOS: IP = &MBR
xxx0	MBR: Load Kernel
xxx1	MBR: IP = &Kernel

xxx0

Processor



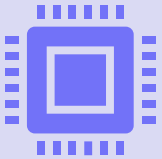
MBR: Load Kernel

Kernel

Address	Memory
0000	BIOS: Check
0001	BIOS: Load MBR
0010	BIOS: IP = &MBR
xxx0	MBR: Load Kernel
xxx1	MBR: IP = &Kernel
yyy0	Kernel: Device Manager
yy11	Kernel: Memory Manager
y111	Kernel: Process Manager

yyy0

Processor



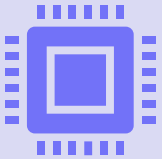
MBR: IP = &Kernel

Kernel

Address	Memory
0000	BIOS: Check
0001	BIOS: Load MBR
0010	BIOS: IP = &MBR
xxx0	MBR: Load Kernel
xxx1	MBR: IP = &Kernel
yyy0	Kernel: Device Manager
yy11	Kernel: Memory Manager
y111	Kernel: Process Manager

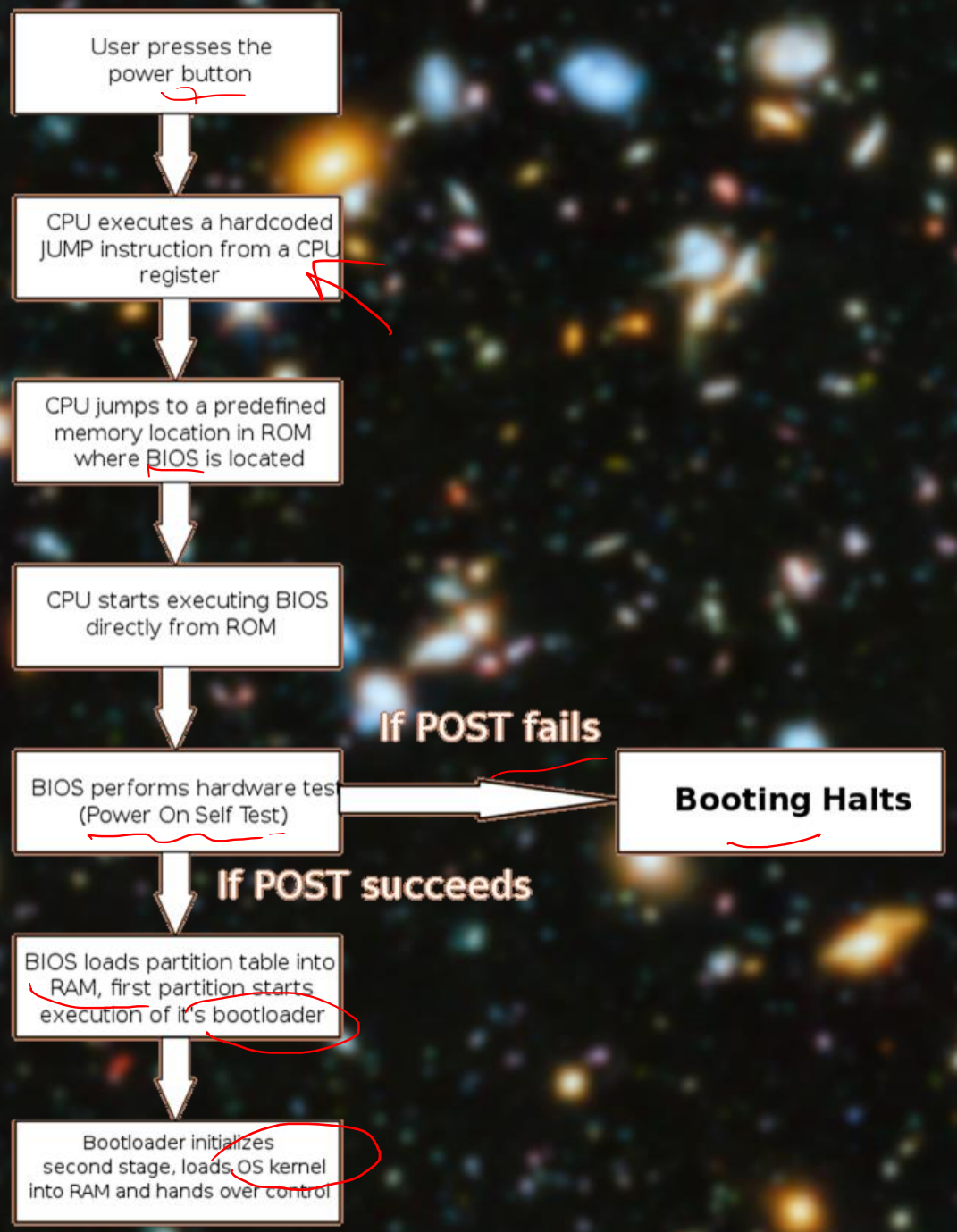
yyy0

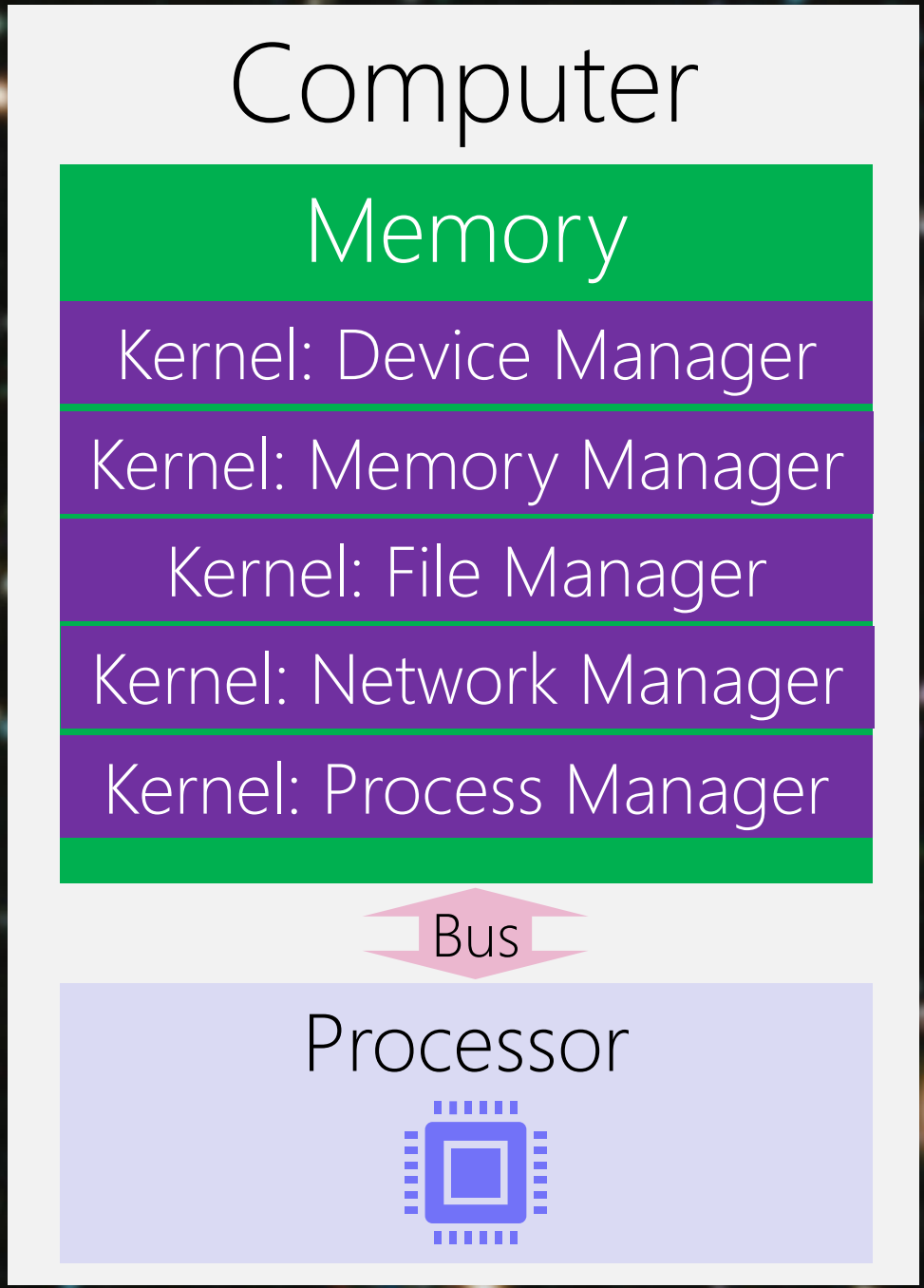
Processor

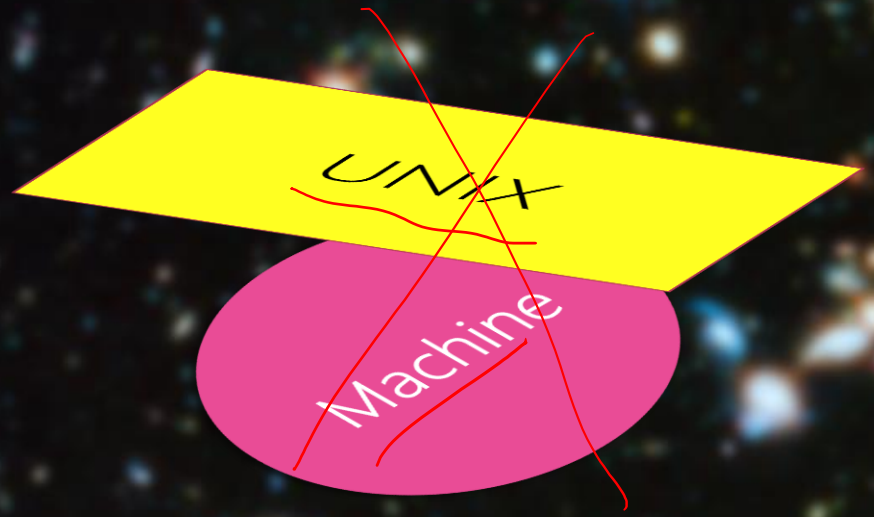


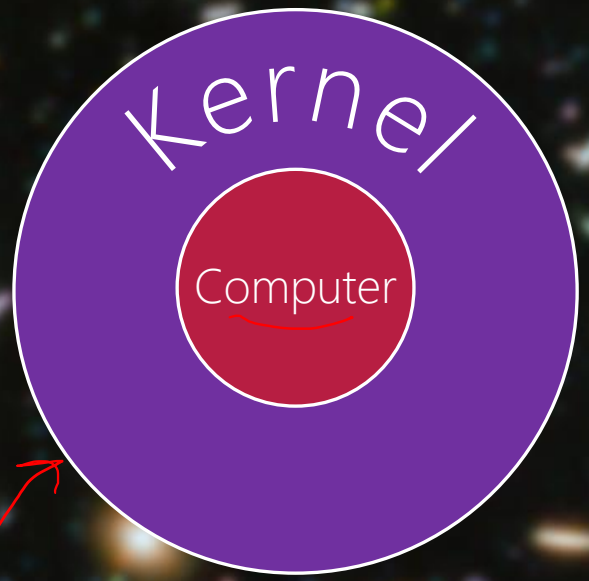
Kernel

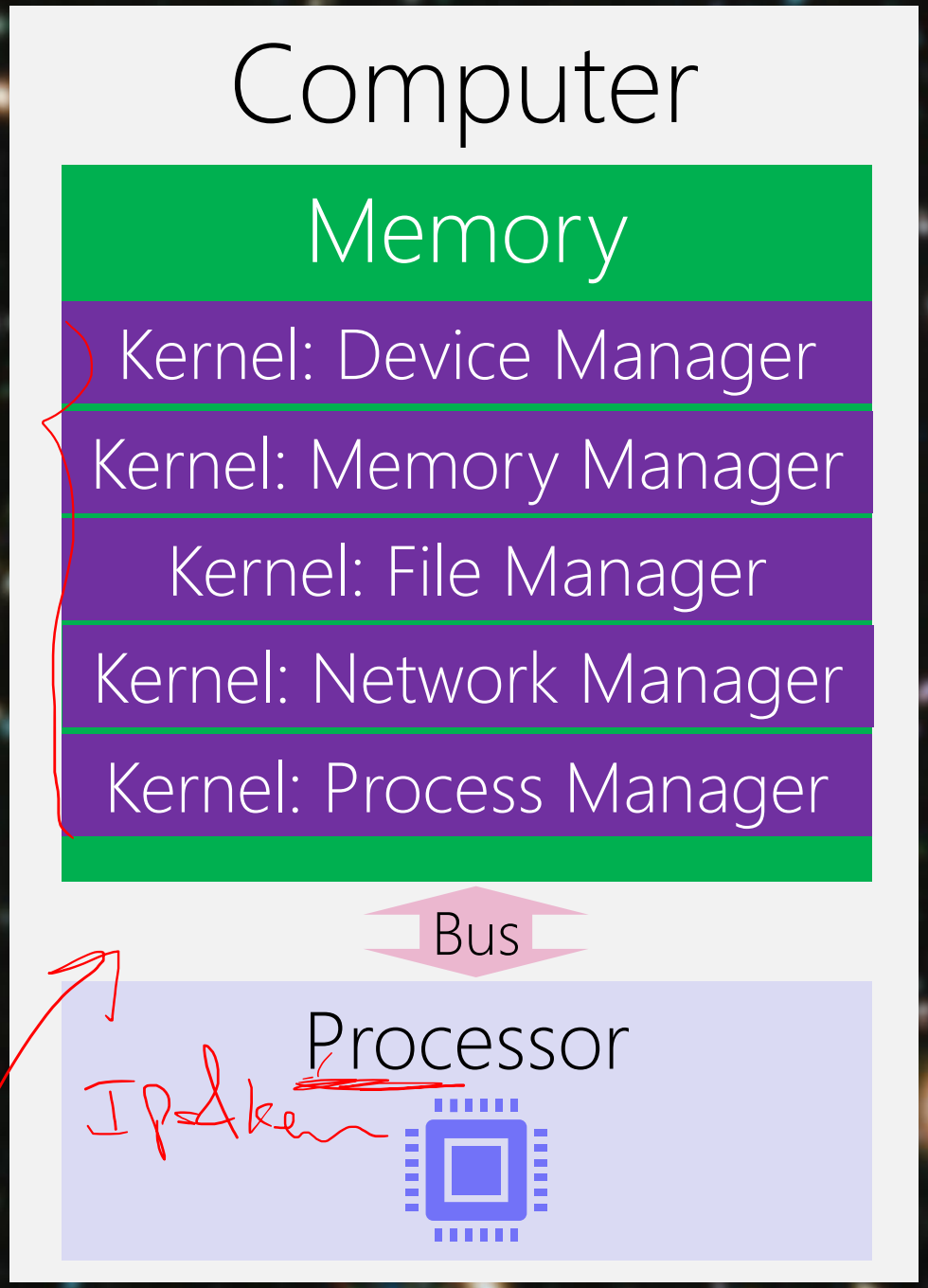
Why not giving the &kernel to bios? Why MBR?!


User presses the power button
CPU executes a hardcoded JUMP instruction from a CPU register
CPU jumps to a predefined memory location in ROM where BIOS is located
CPU starts executing BIOS directly from ROM
BIOS performs hardware test (Power On Self Test)
If POST fails
Booting Halts
If POST succeeds
BIOS loads partition table into RAM, first partition starts execution of it's bootloader
Bootloader initializes second stage, loads OS kernel into RAM and hands over control


Computer
Memory
Kernel: Device Manager
Kernel: Memory Manager
Kernel: File Manager
Kernel: Network Manager
Kernel: Process Manager
Bus
Processor


UNIX
Machine

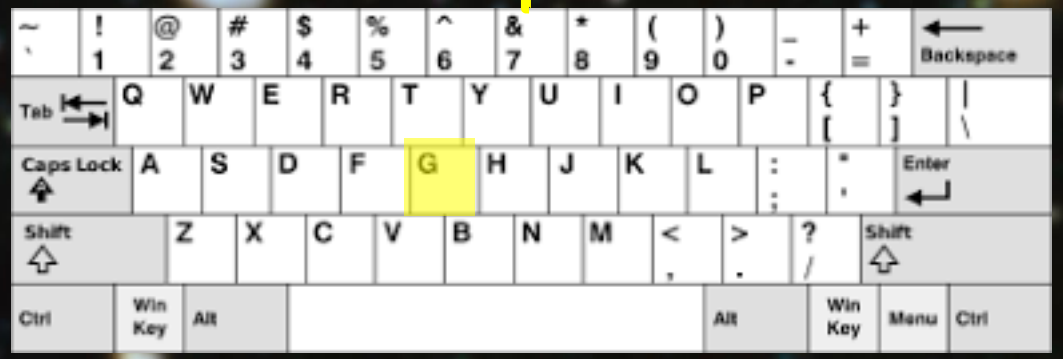

Kernel
Computer

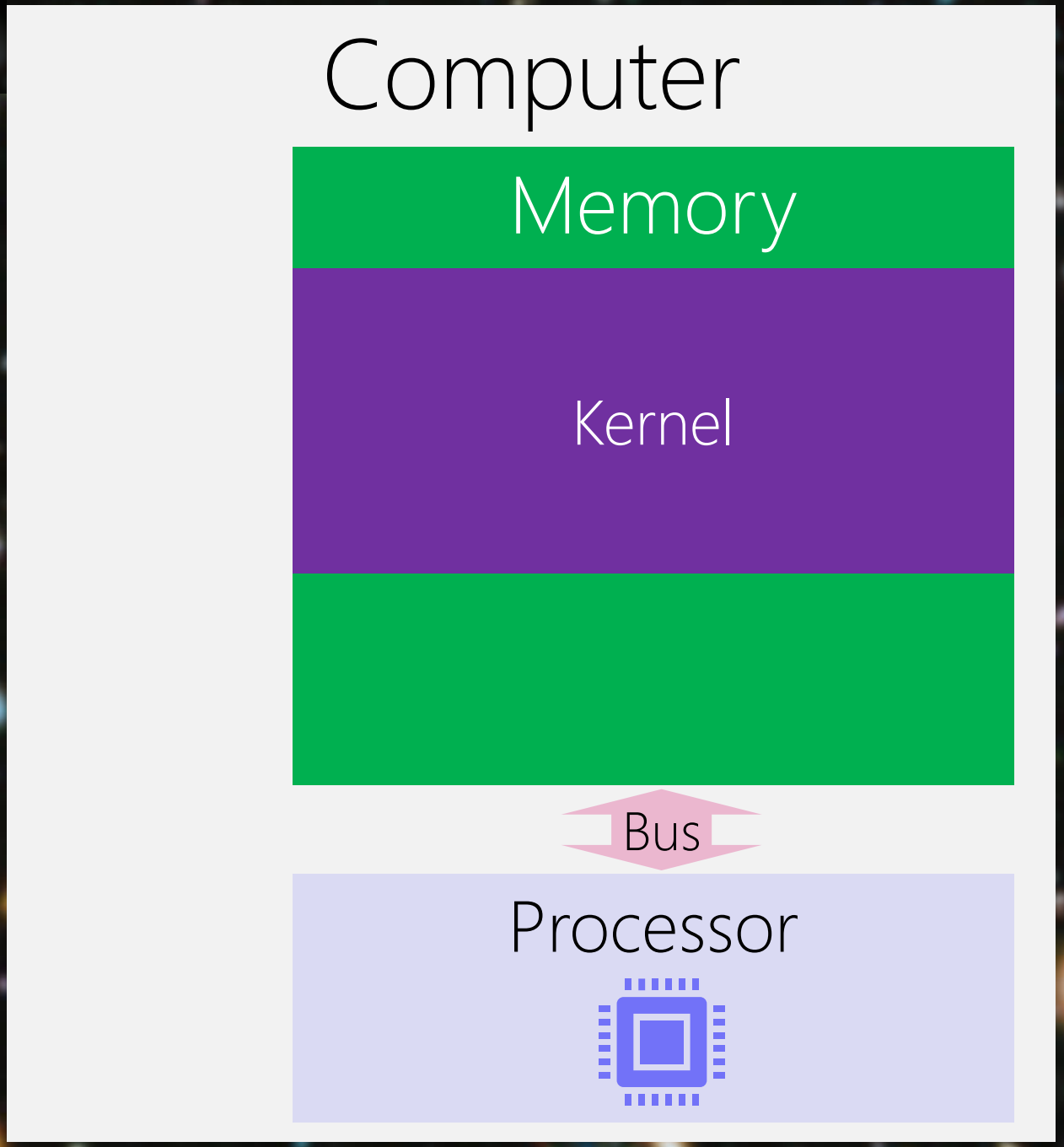

Computer
Memory
Kernel: Device Manager
Kernel: Memory Manager
Kernel: File Manager
Kernel: Network Manager
Kernel: Process Manager
Bus
Processor

Keystroke

mouse click, usb plug, graphic show, ...

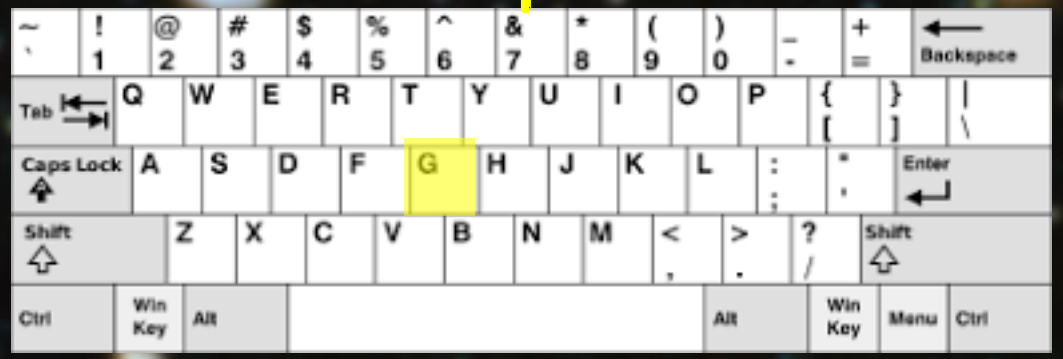
E.g., Keystroke

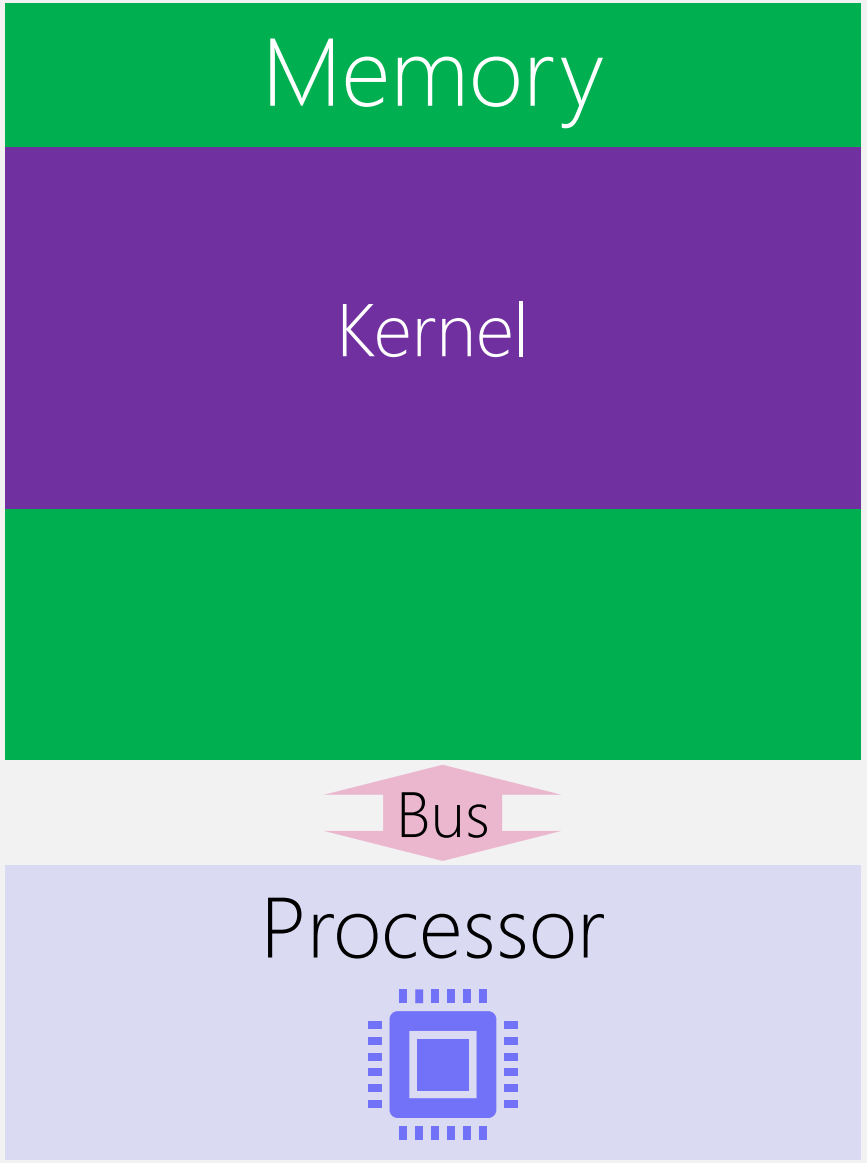




E.g., Keystroke

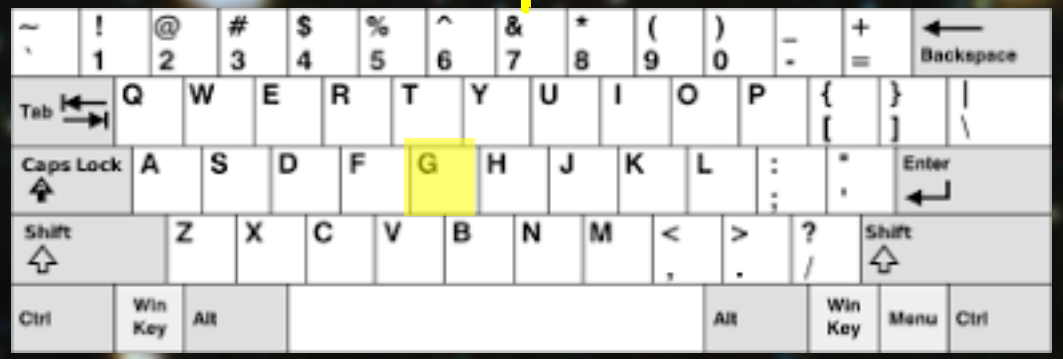
1. Electric Signal

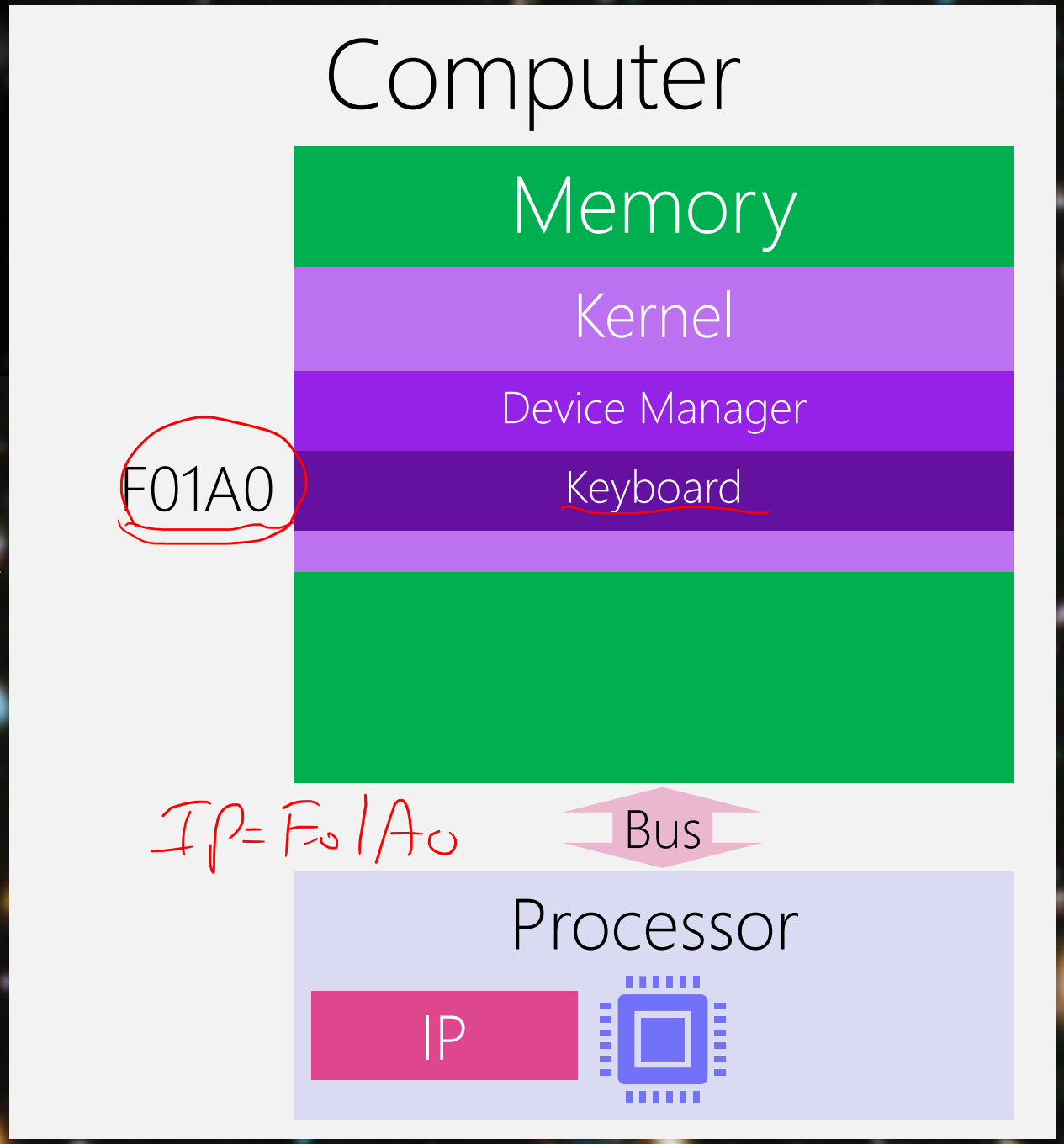




E.g., Keystroke

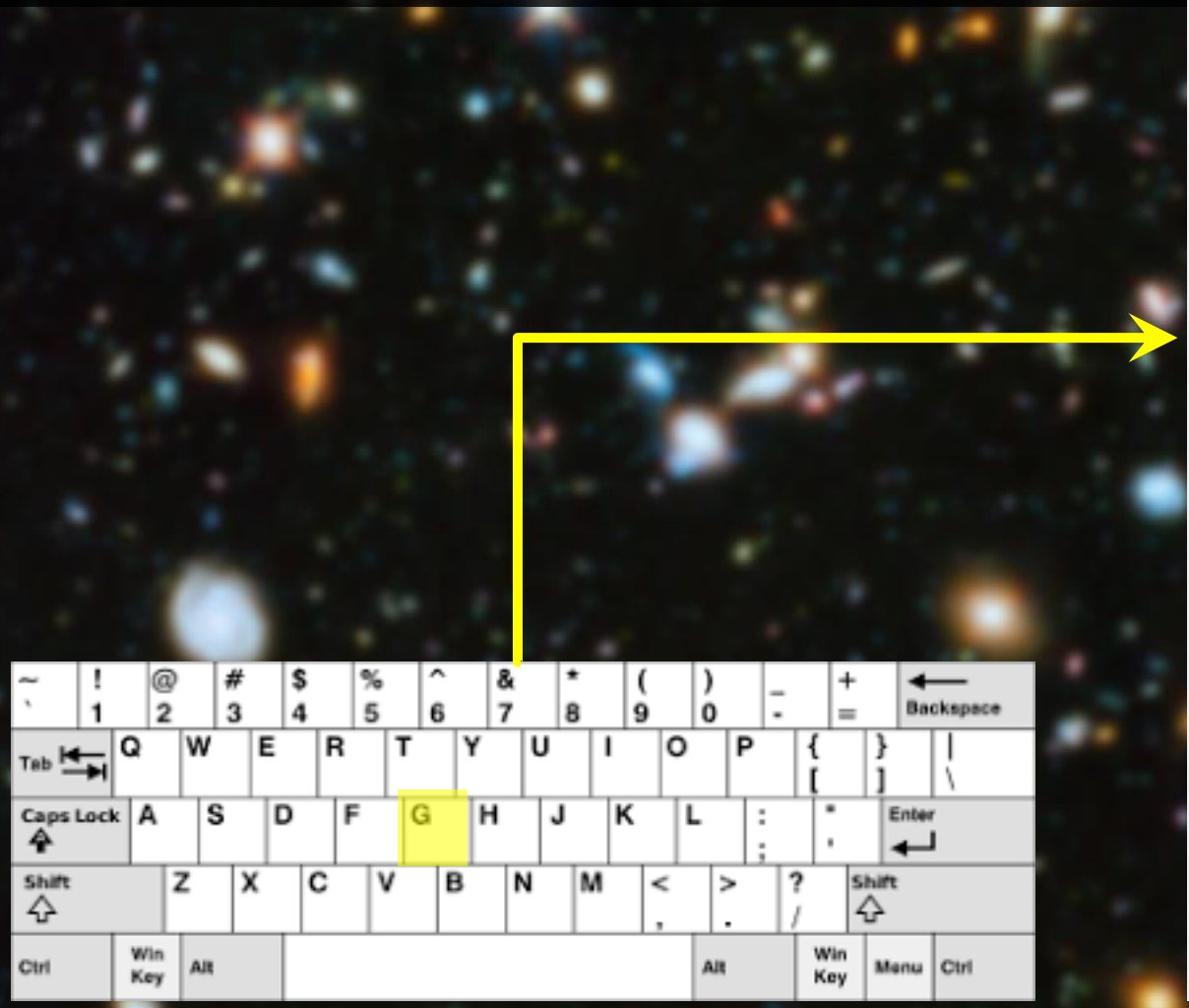
2. Kernel's Device Manager for Keyboard

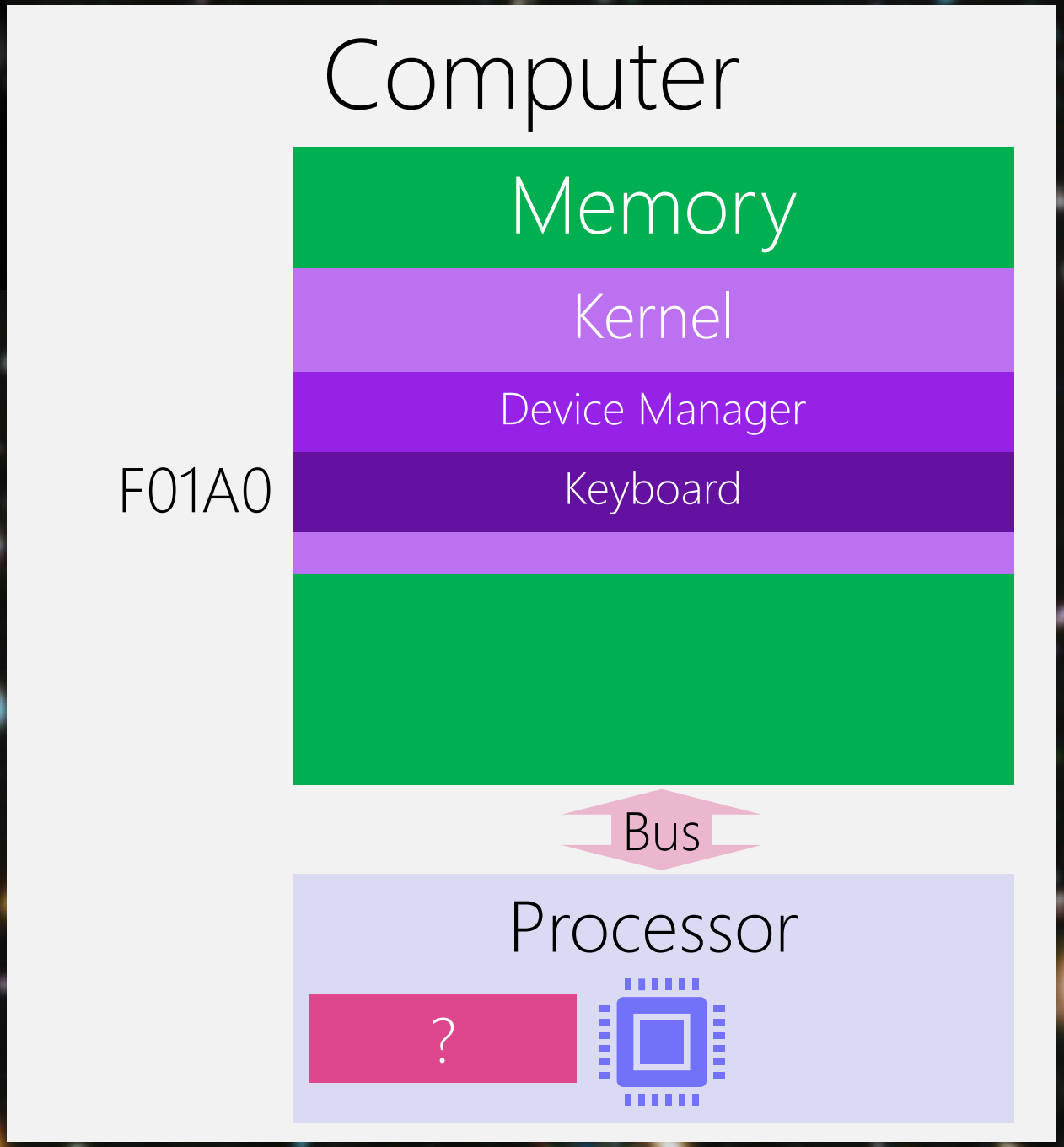




E.g., Keystroke

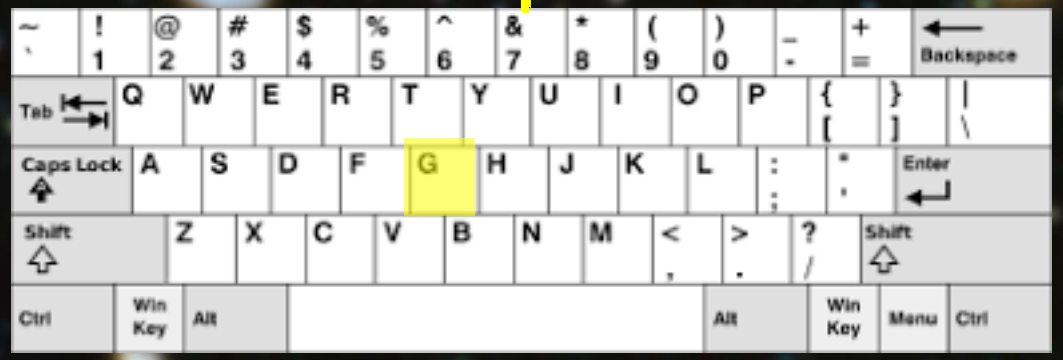
3. IP = F01A0

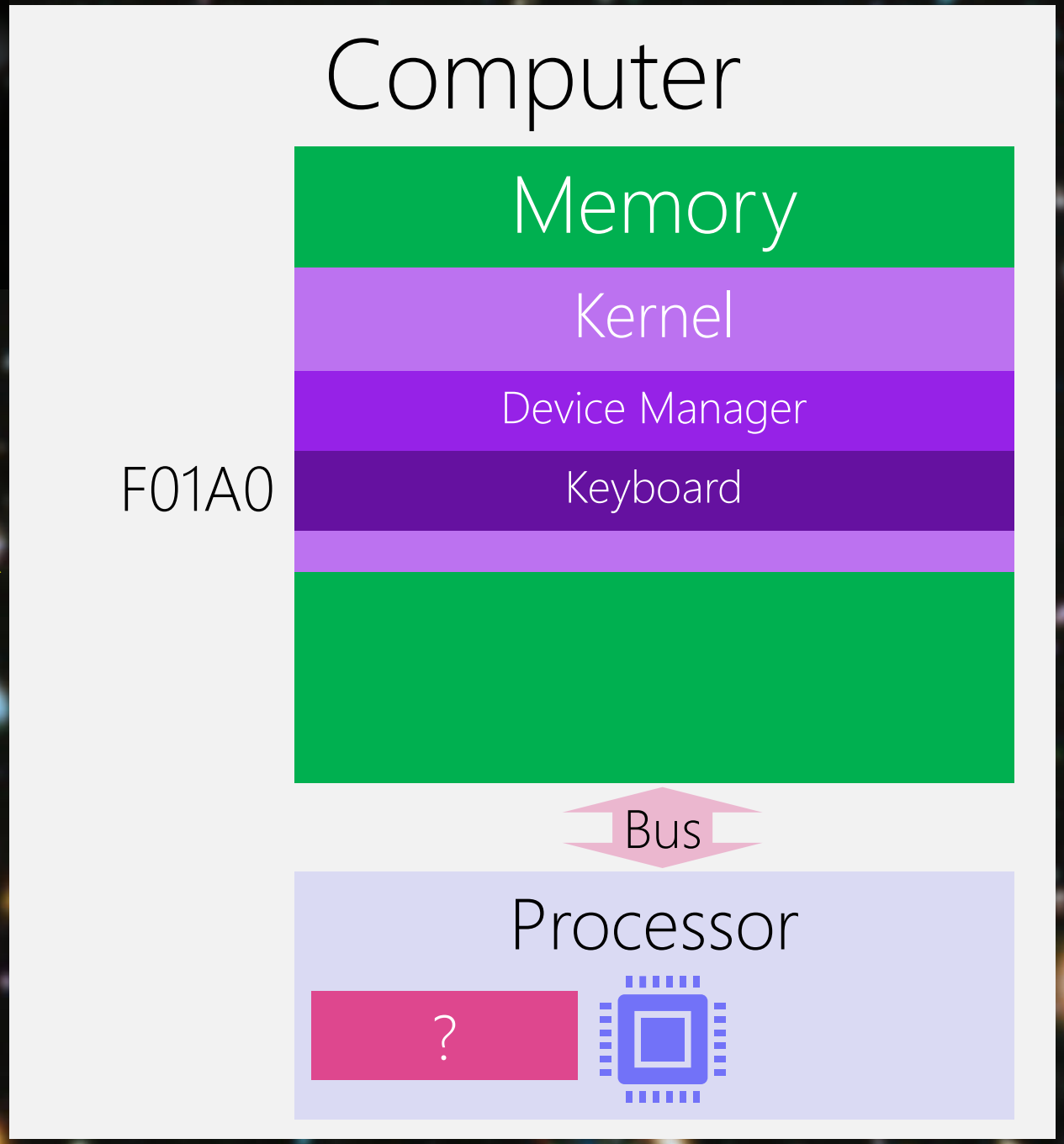




E.g., Keystroke

Who does IP = F01A0?

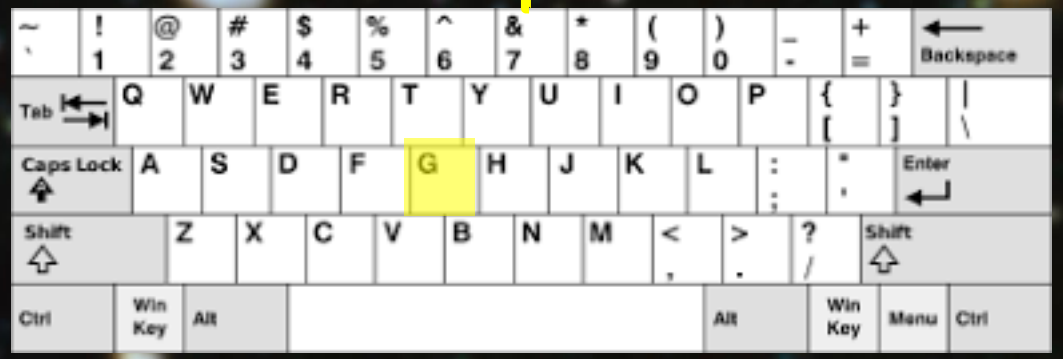


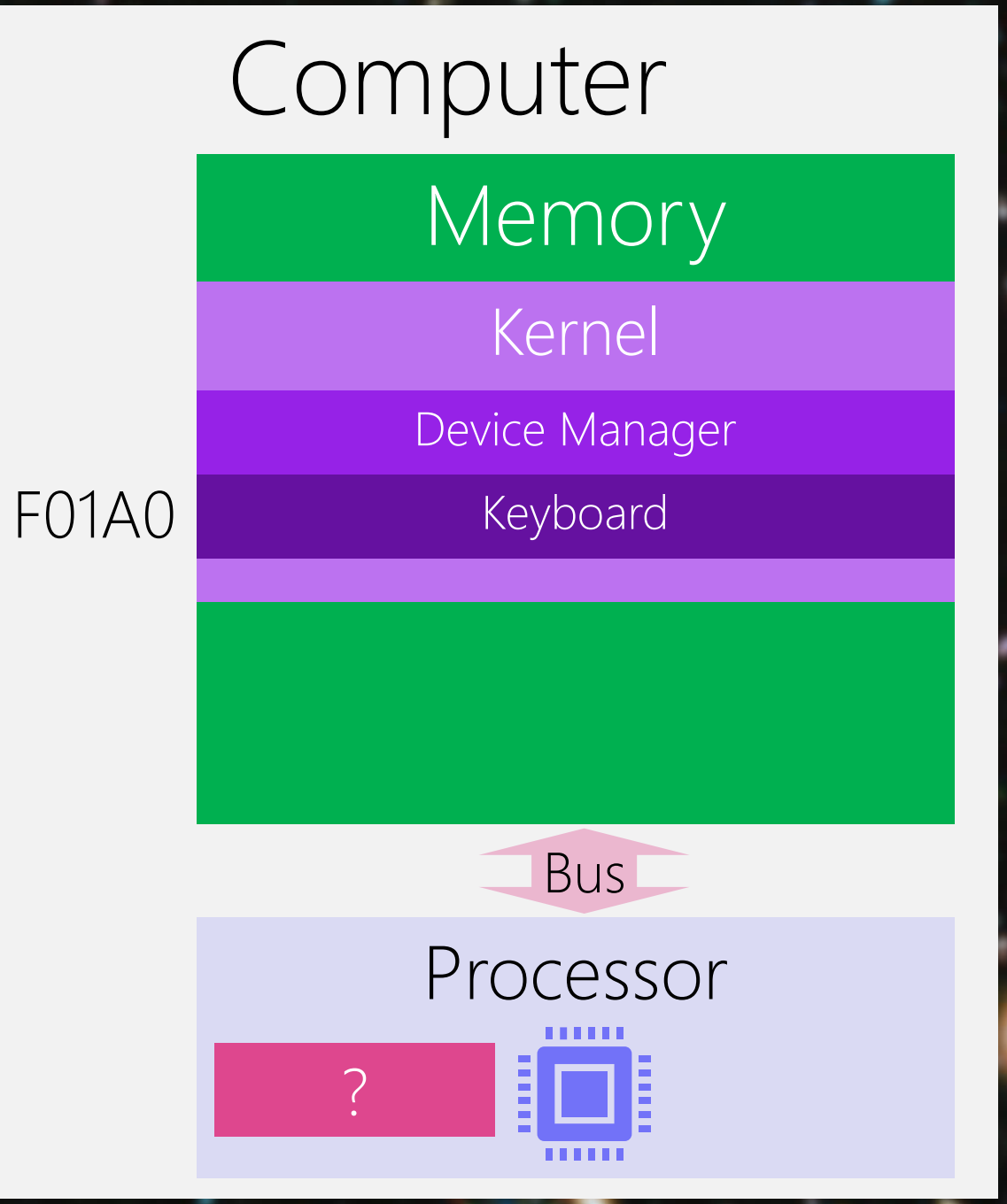


E.g., Keystroke

Who does IP = F01A0?

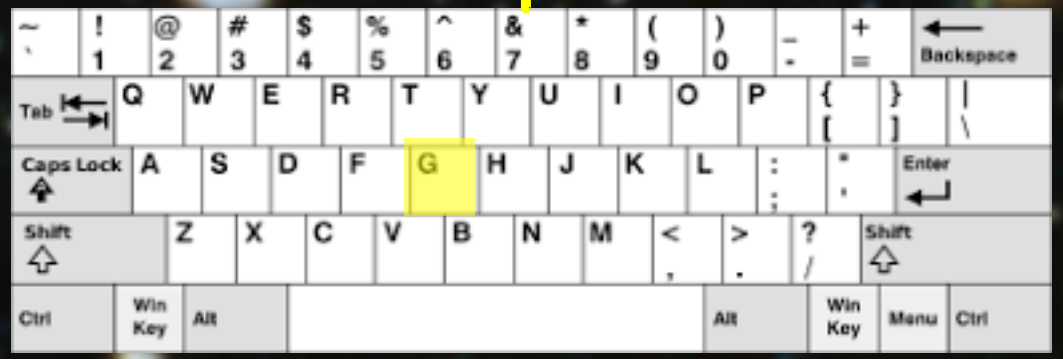
Electrical Signal → Kernel's Program

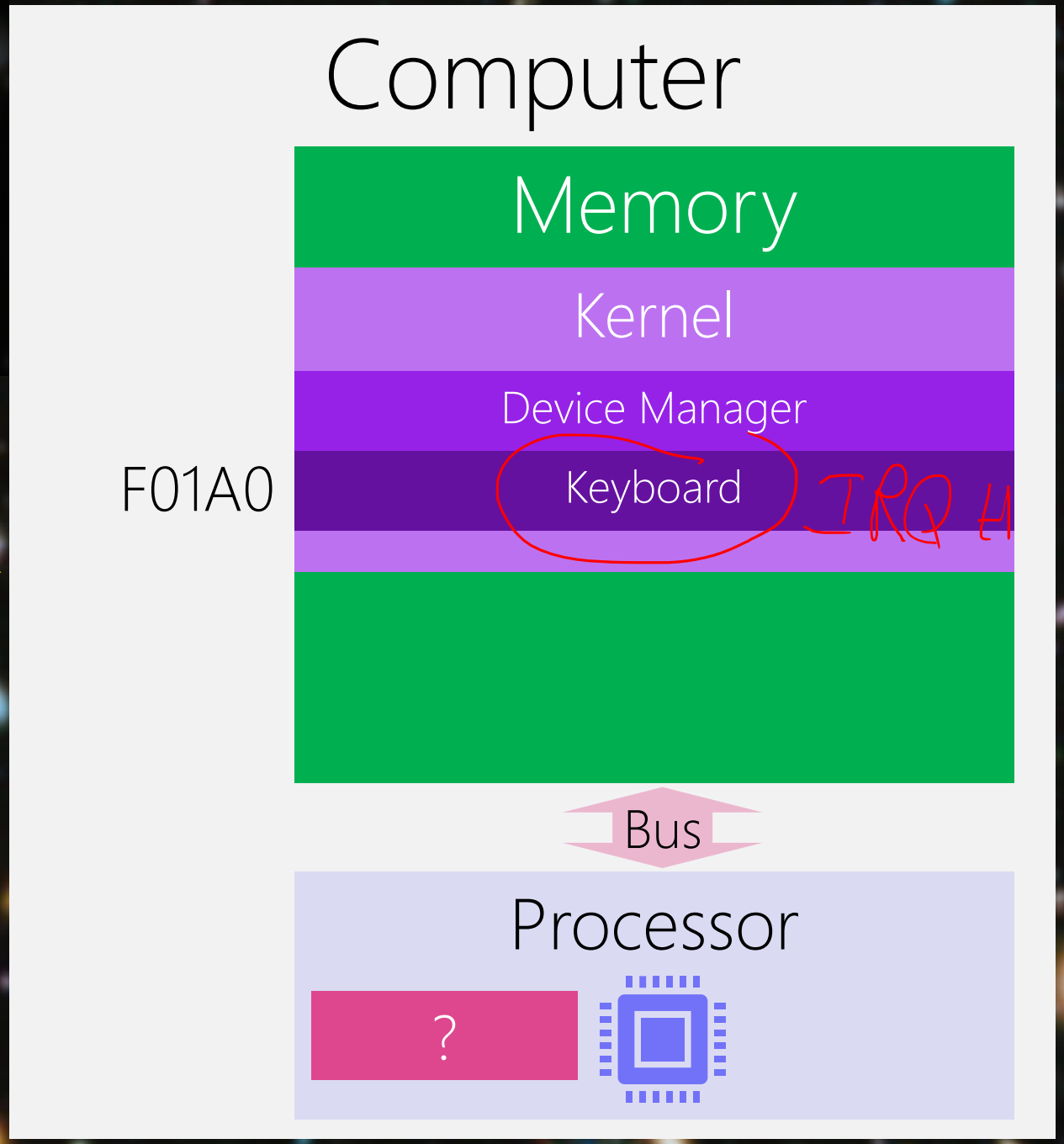




E.g., Keystroke

Interrupt Request (IRQ)
Interrupt Request Handler





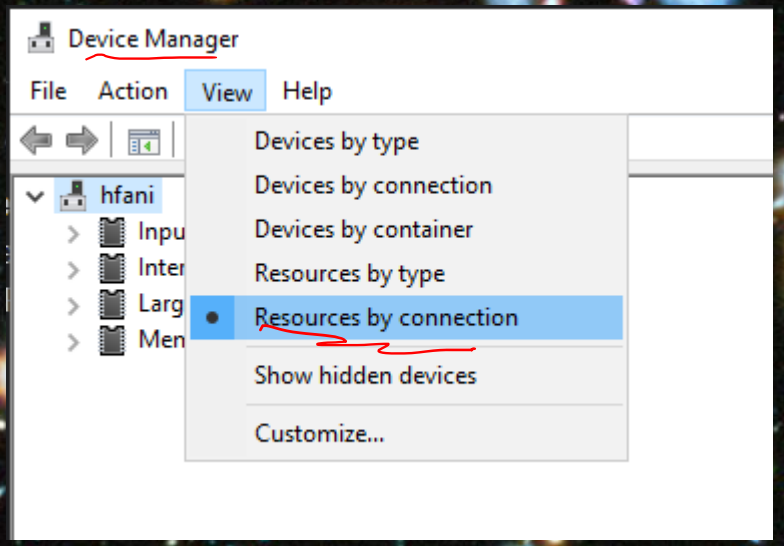
```
bob@suse1:~> sudo cat /proc/interrupts
           CPU0
  0:         78   IO-APIC-edge      timer
  1:       9012   IO-APIC-edge      i8042
  3:          1   IO-APIC-edge
  4:       6609   IO-APIC-edge
  6:          7   IO-APIC-edge      floppy
  7:          0   IO-APIC-edge      parport0
  8:          1   IO-APIC-edge      rtc0
  9:          0   IO-APIC-fasteoi   acpi
 12:     101705   IO-APIC-edge      i8042
 14:       1374   IO-APIC-edge      ata_piix
 15:      28050   IO-APIC-edge      ata_piix
 16:        893   IO-APIC-fasteoi   Ensoniq AudioPCI
 17:      77088   IO-APIC-fasteoi   ioc0, ehci_hcd:usb1
 18:        112   IO-APIC-fasteoi   uhci_hcd:usb2
 19:       5167   IO-APIC-fasteoi   eth0
 40:          0   PCI-MSI-edge      PCIe PME, pciehp
 41:          0   PCI-MSI-edge      PCIe PME, pciehp
 42:          0   PCI-MSI-edge      PCIe PME, pciehp
 43:          0   PCI-MSI-edge      PCIe PME, pciehp
 44:          0   PCI-MSI-edge      PCIe PME, pciehp
 45:          0   PCI-MSI-edge      PCIe PME, pciehp
 46:          0   PCI-MSI-edge      PCIe PME, pciehp
```

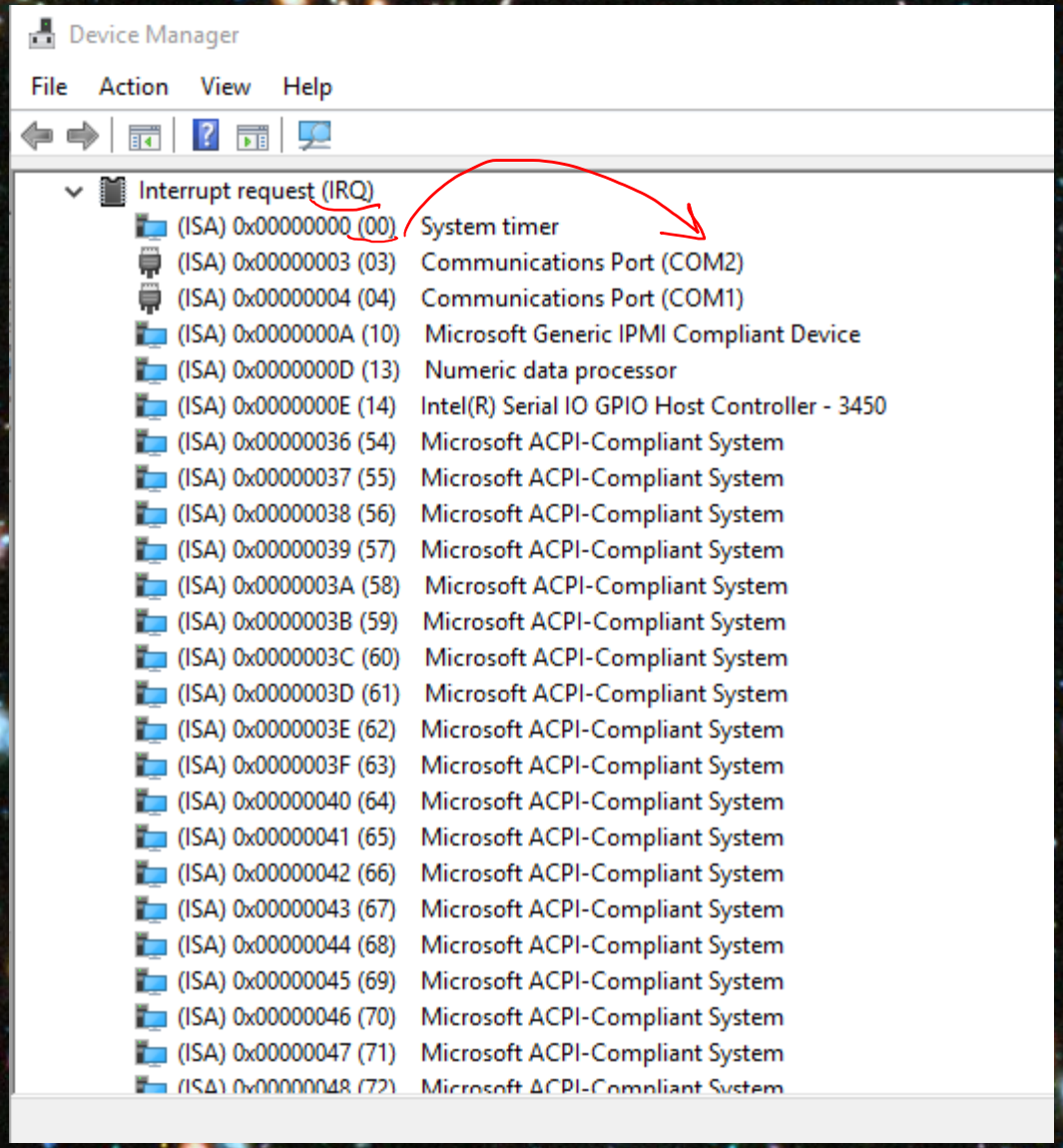
010

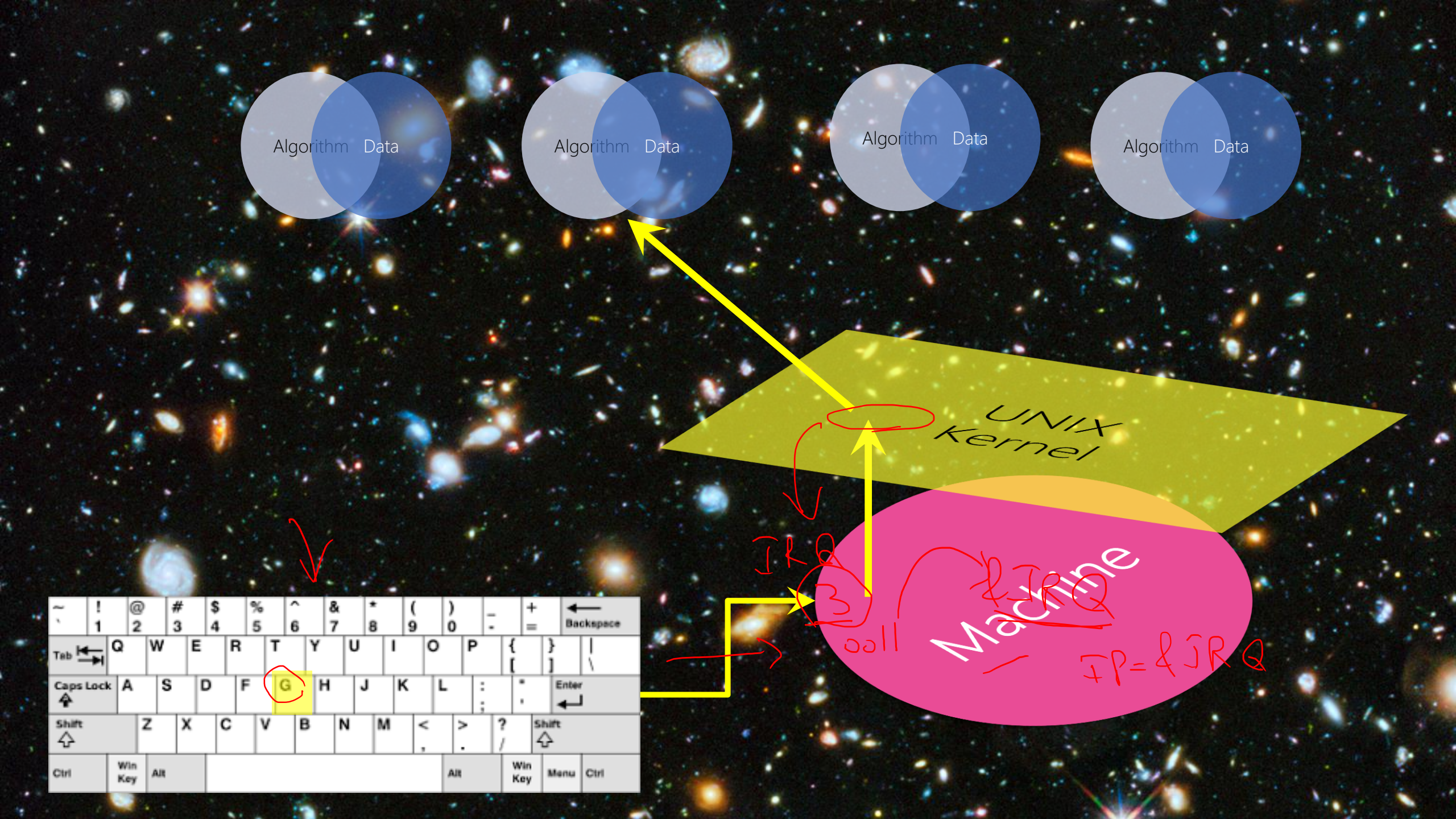
IRQ	Usage
0	system timer (cannot be changed)
1	keyboard controller (cannot be changed)
2	cascaded signals from IRQs 8–15
3	second RS-232 serial port (COM2: in Windows)
4	first RS-232 serial port (COM1: in Windows)
5	parallel port 2 and 3 or sound card
6	floppy disk controller
7	first parallel port
8	real-time clock
9	open interrupt
10	open interrupt
11	open interrupt
12	PS/2 mouse
13	math coprocessor
14	primary ATA channel
15	secondary ATA channel

```
hfani@charlie:~$ procinfo
Memory:          Total        Used        Free      Buffer
RAM:        264118524    22664004   241454520      105852
Swap:        67378172           0    67378172

Bootup: Thu Aug 19 15:26:13 2021   Load average: 0.17 0

user  :      1w 1d 12:02:18.71   5.3%  page in :
nice  :           00:33:45.08   0.0%  page out:
system:      3w 2d 02:48:23.42  14.5%  page act:
IOwait:           01:10:09.71   0.0%  page dea:
hw irq:           00:00:00.00   0.0%  page flt:        3
sw irq:           09:31:11.65   0.2%  swap in :
idle  :     18w 1d 12:22:16.05  79.9%  swap out:
uptime:      1w 2d 23:19:40.99         context :      2961053551

irq  0:          49  2-edge timer        irq  33:    3636919  1572868-edge eth0
irq  1:           4  1-edge i8042        irq  34:   12534197  1572869-edge eth0
irq  8:           1  8-edge rtc0         irq  35:    3728604  1572870-edge eth0
irq  9:           0  9-fasteoi acpi      irq  36:    6064430  1572871-edge eth0
irq 12:           6  12-edge i8042       irq  37:    5535547  1572872-edge eth0
irq 14:           0  14-edge pata_atii   irq  39:    6264652  1048576-edge mega
irq 15:           0  15-edge pata_atii   irq  41:          0  1574912-edge eth1
irq 16:           6  16-fasteoi ohci_h   irq  42:     430781  1574913-edge eth1
irq 17:           0  17-fasteoi ehci_h   irq  43:     430781  1574914-edge eth1
irq 18:          37  18-fasteoi ohci_h   irq  44:     430781  1574915-edge eth1
irq 19:           2  19-fasteoi ehci_h   irq  45:     430781  1574916-edge eth1
```

IRQ	Usage
0	system timer (cannot be changed)
1	keyboard controller (cannot be changed)
2	cascaded signals from IRQs 8–15
3	second RS-232 serial port (COM2: in Windows)
4	first RS-232 serial port (COM1: in Windows)
5	parallel port 2 and 3 or sound card
6	floppy disk controller
7	first parallel port
8	real-time clock
9	open interrupt
10	open interrupt
11	open interrupt
12	PS/2 mouse
13	math coprocessor
14	primary ATA channel
15	secondary ATA channel


Device Manager
File
Action
View
Help
hfani
Inpu
Inter
Larg
Men
Devices by type
Devices by connection
Devices by container
Resources by type
Resources by connection
Show hidden devices
Customize...

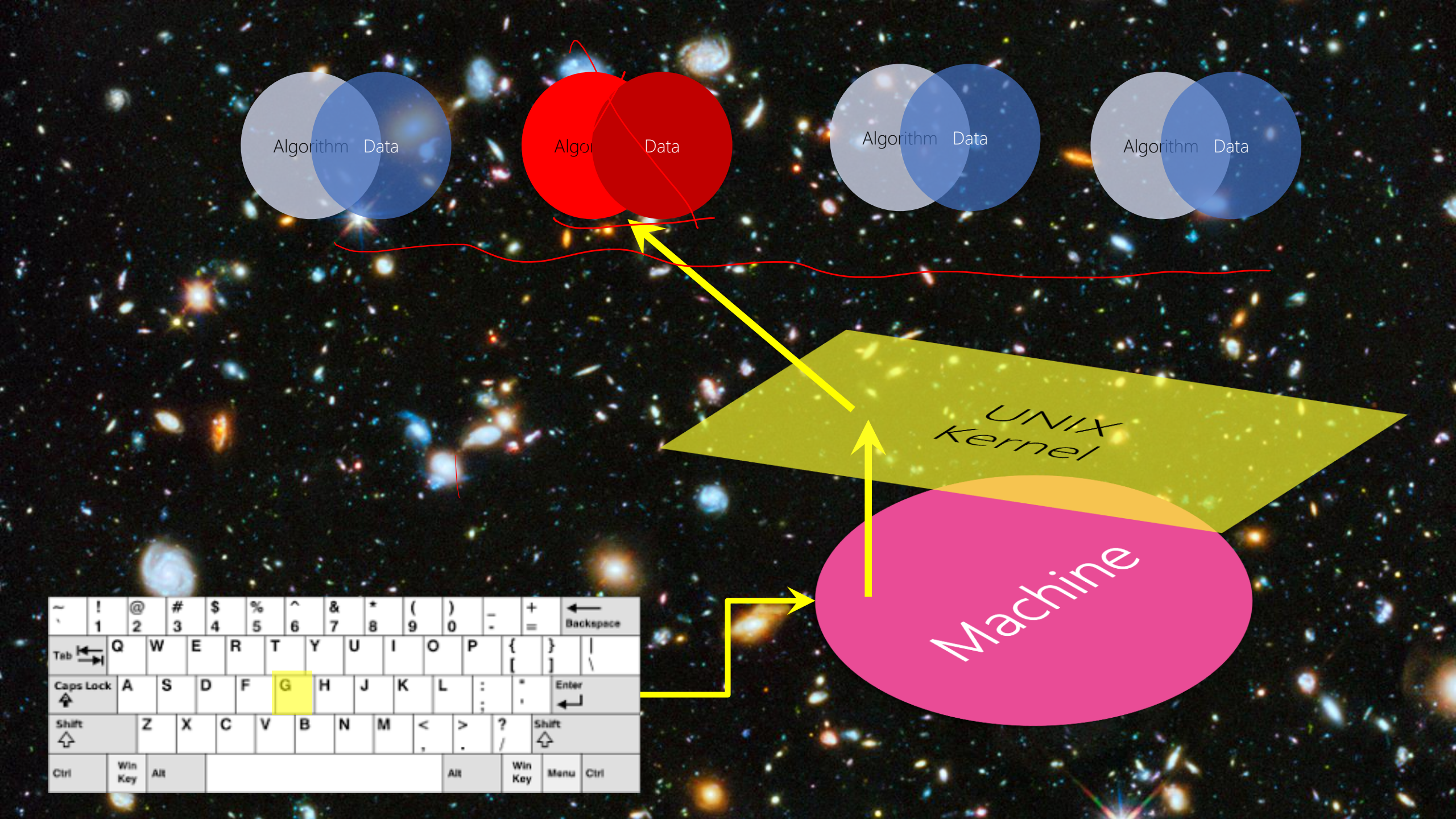

Device Manager
File
Action
View
Help
Interrupt request (IRQ)
(ISA) 0x00000000 (00) System timer
(ISA) 0x00000003 (03) Communications Port (COM2)
(ISA) 0x00000004 (04) Communications Port (COM1)
(ISA) 0x0000000A (10) Microsoft Generic IPMI Compliant Device
(ISA) 0x0000000D (13) Numeric data processor
(ISA) 0x0000000E (14) Intel(R) Serial IO GPIO Host Controller - 3450
(ISA) 0x00000036 (54) Microsoft ACPI-Compliant System
(ISA) 0x00000037 (55) Microsoft ACPI-Compliant System
(ISA) 0x00000038 (56) Microsoft ACPI-Compliant System
(ISA) 0x00000039 (57) Microsoft ACPI-Compliant System
(ISA) 0x0000003A (58) Microsoft ACPI-Compliant System
(ISA) 0x0000003B (59) Microsoft ACPI-Compliant System
(ISA) 0x0000003C (60) Microsoft ACPI-Compliant System
(ISA) 0x0000003D (61) Microsoft ACPI-Compliant System
(ISA) 0x0000003E (62) Microsoft ACPI-Compliant System
(ISA) 0x0000003F (63) Microsoft ACPI-Compliant System
(ISA) 0x00000040 (64) Microsoft ACPI-Compliant System
(ISA) 0x00000041 (65) Microsoft ACPI-Compliant System
(ISA) 0x00000042 (66) Microsoft ACPI-Compliant System
(ISA) 0x00000043 (67) Microsoft ACPI-Compliant System
(ISA) 0x00000044 (68) Microsoft ACPI-Compliant System
(ISA) 0x00000045 (69) Microsoft ACPI-Compliant System
(ISA) 0x00000046 (70) Microsoft ACPI-Compliant System
(ISA) 0x00000047 (71) Microsoft ACPI-Compliant System
(ISA) 0x00000048 (72) Microsoft ACPI-Compliant System

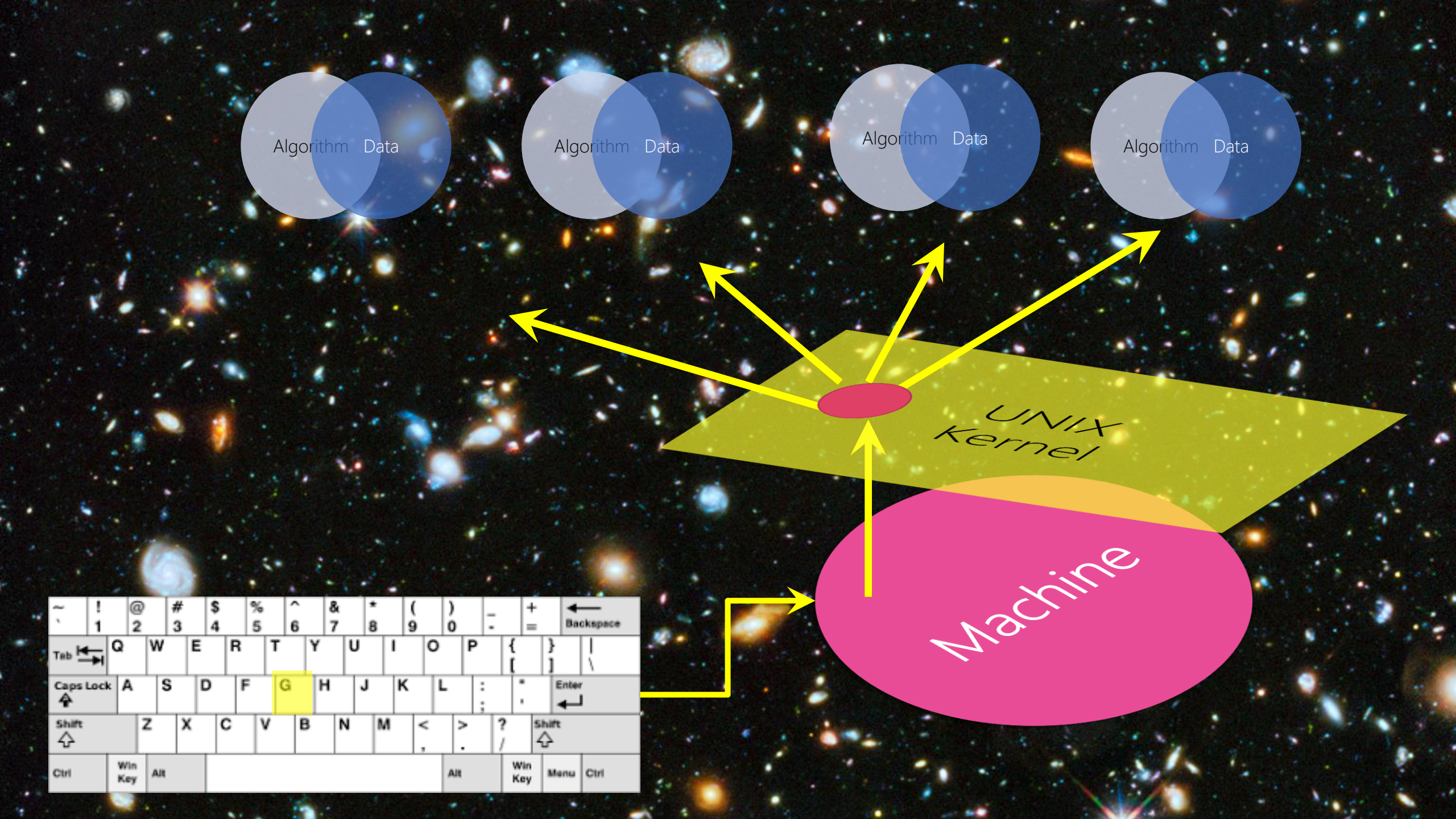

Algorithm
Data
Algorithm
Data
Algorithm
Data
Algorithm
Data
UNIX
Kernel
Machine
IRQ
&IRQ
IP=&IRQ
Backspace
Tab
Caps Lock
Enter
Shift
Shift
Ctrl
Win Key
Alt
Alt
Win Key
Menu
Ctrl

Keystroke Hack

keylogger

Application-level vs. System-level


Algorithm
Data
Algorithm
Data
Algorithm
Data
Algorithm
Data
UNIX
Kernel
Machine


Algorithm
Data
Algorithm
Data
Algorithm
Data
Algorithm
Data
UNIX
Kernel
Machine

Keystroke Hack

Shutdown

Application-level vs. System-level

0 → timer

Who sets the IRQ table?

Machine? BIOS? UNIX (Kernel)?

Program → Process


Ezekiel 25:17, Pulp Fiction (1994), Quentin Tarantino
You read the Bible, Brett?


Ezekiel 25:17, Pulp Fiction (1994), Quentin Tarantino
https://www.youtube.com/watch?v=x2WK_eWihdU
Sensitive Content
This photo contains sensitive content which some people may find offensive or disturbing.
Well, there's this passage I've got memorized; sorta fits this occasion.

Program → Process

Hello World!

"Then the LORD God formed a man from the dust of the ground and breathed into his nostrils the breath of life, and the man became a living being." – 'Genesis 2:7

C

```
hfani@alpha:~$ vi hello.c
hfani@alpha:~$ cc main.c -S
```

```c
#include <stdio.h>
void main(){
    printf("hello world!");
}
```

Compiler

Assembly

```
hfani@alpha:~$ cc main.s -c
```

```
<printf@plt>:
jmpq        *0x3002(%rip)
pushq       $0x0
jmpq        401000 <.plt>
<main>:
push        %rbp
mov         %rsp,%rbp
lea         0xfd5(%rip),%rdi
mov         $0x0,%eax
callq       401010 <printf@plt>
nop
pop         %rbp
retq
```

Assembler

OP Code

```
hfani@alpha:~$ cc main.o -o main
```

```
0001 0000 0000 0000 0000 0000 0000 0000
0011 0000 0003 0000 0000 0000 0000 0000
0000 0000 0000 0000 3309 0000 0000 0000
0096 0000 0000 0000 0000 0000 0000 0000
0001 0000 0000 0000 0000 0000 0000 0000
```

Program

```
0001 0000 0000 0000 0000 0000 0000 0000
0011 0000 0003 0000 0000 0000 0000 0000
0000 0000 0000 0000 3309 0000 0000 0000
0096 0000 0000 0000 0000 0000 0000 0000
0001 0000 0000 0000 0000 0000 0000 0000
```

Computer

Memory to Store

Kernel

FF1C0

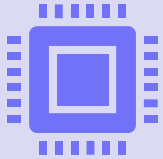
Process Manager
Program → Process

Bus

IP=~~&~~FF1C0

Processor

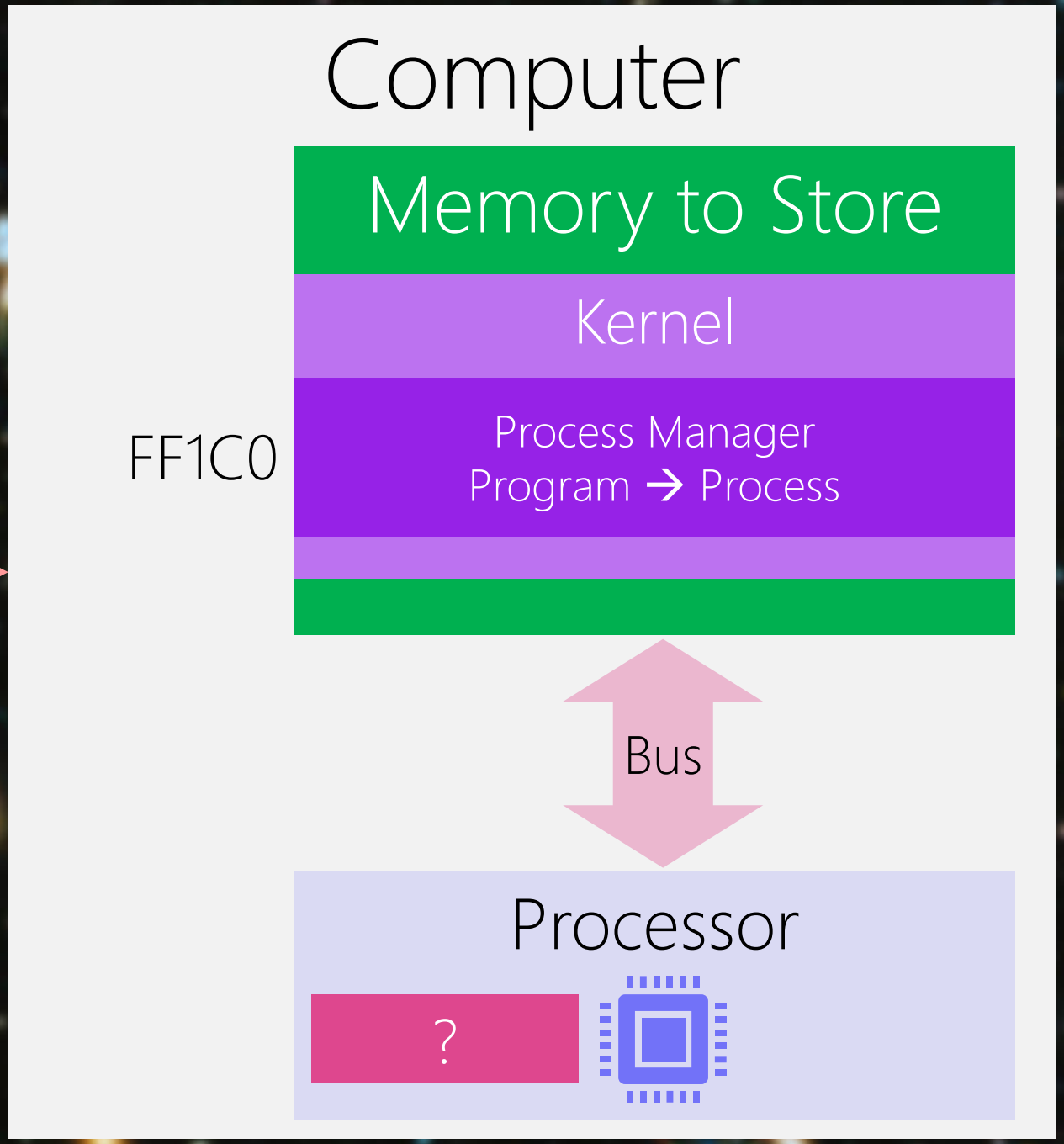
?



Program

```
0001 0000 0000 0000 0000 0000 0000 0000
0011 0000 0003 0000 0000 0000 0000 0000
0000 0000 0000 0000 3309 0000 0000 0000
0096 0000 0000 0000 0000 0000 0000 0000
0001 0000 0000 0000 0000 0000 0000 0000
```

1. Locate the program: /home/hfani/...
2. Call Process Manager for loading the program
   IP = FF1C0
3. Point IP to the first line of the program
   IP = &first opcode



SHELL

Application-level program to act as a Dispatcher