

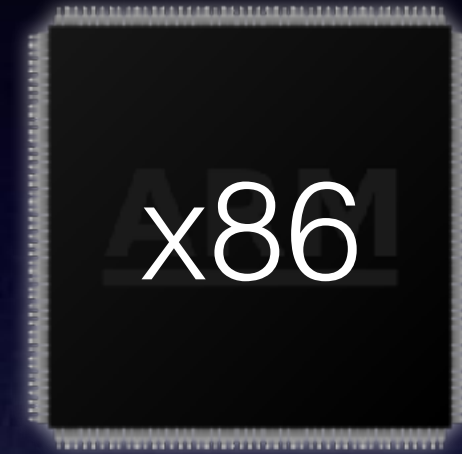


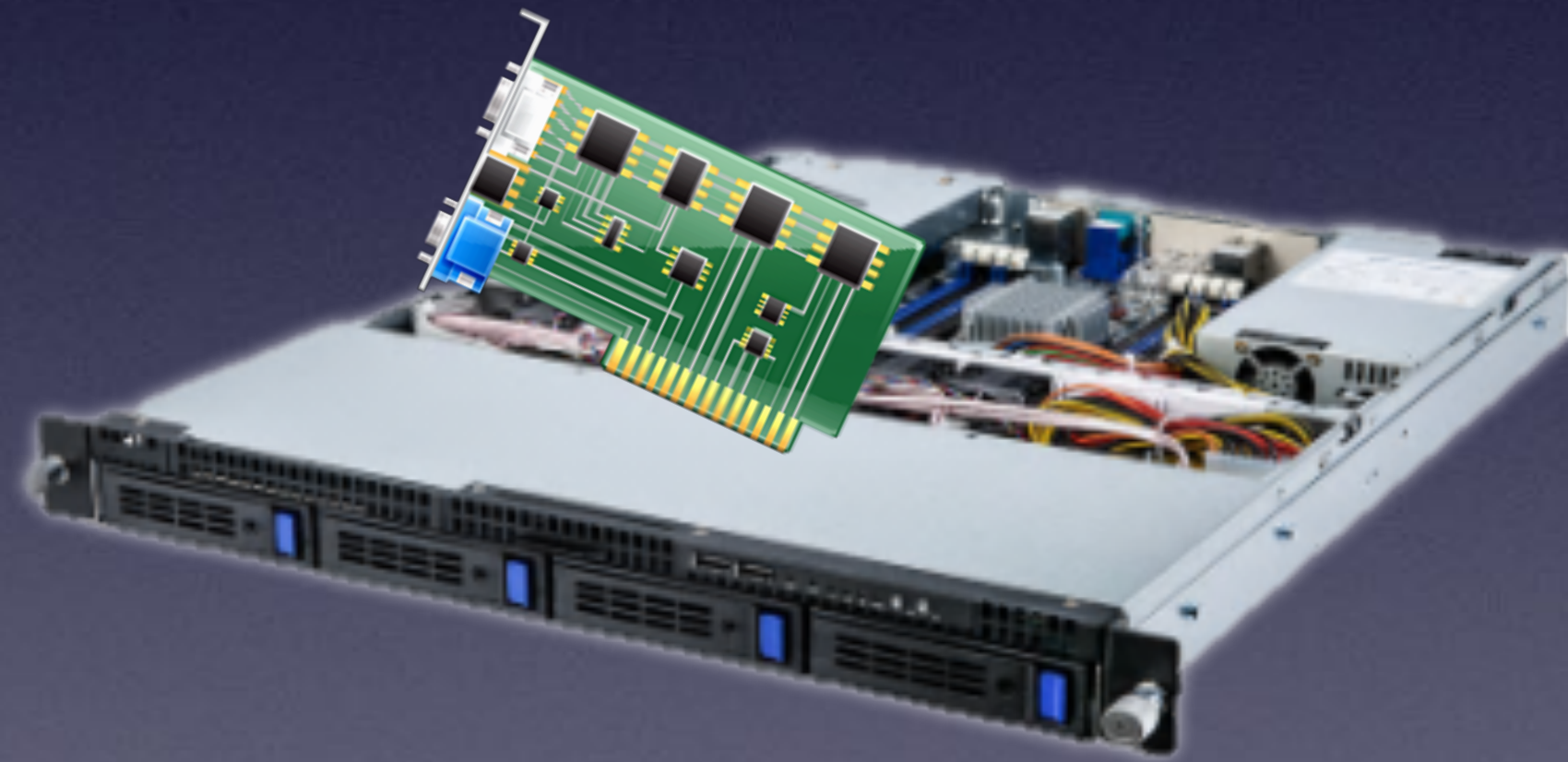
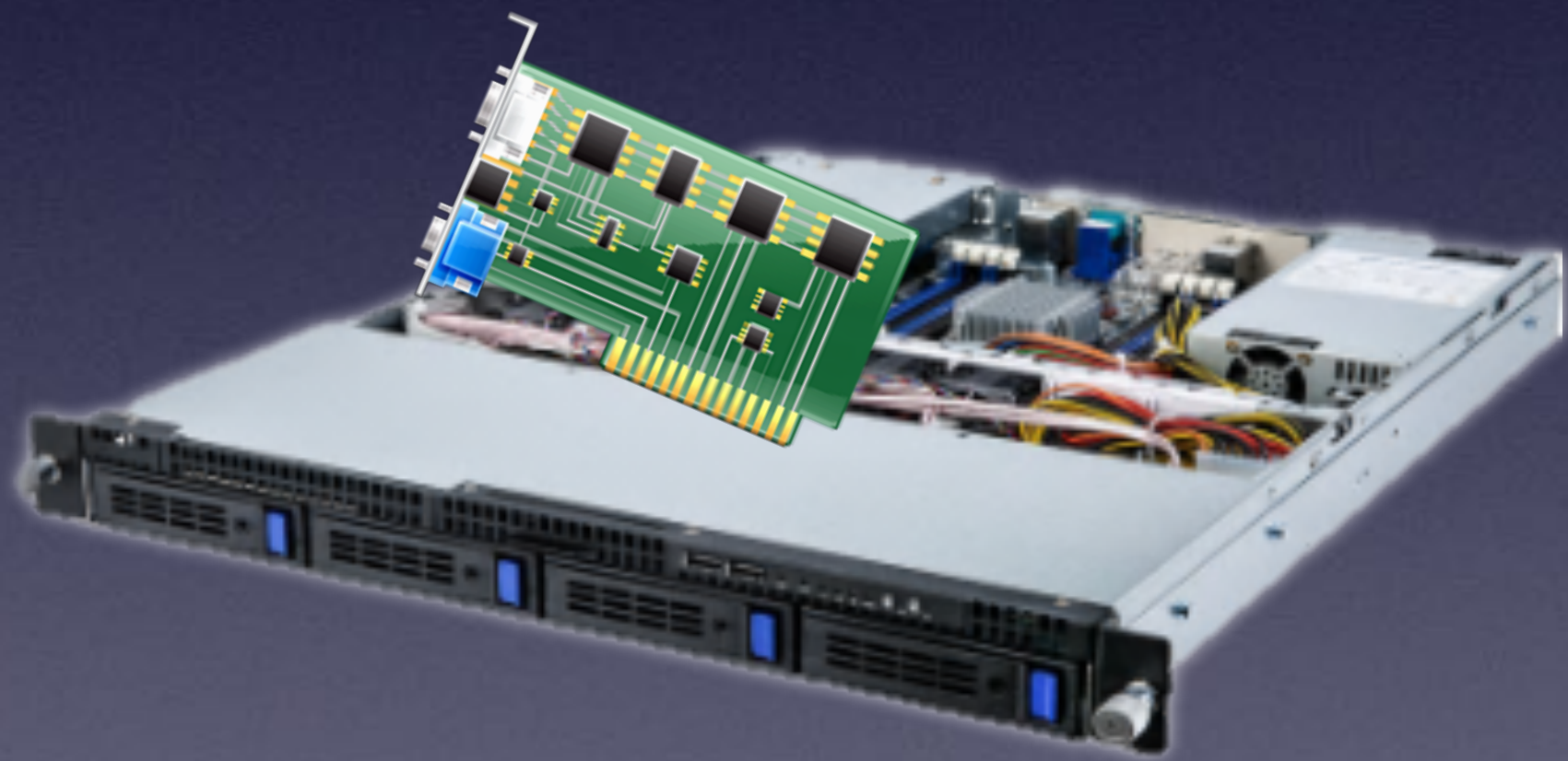
in

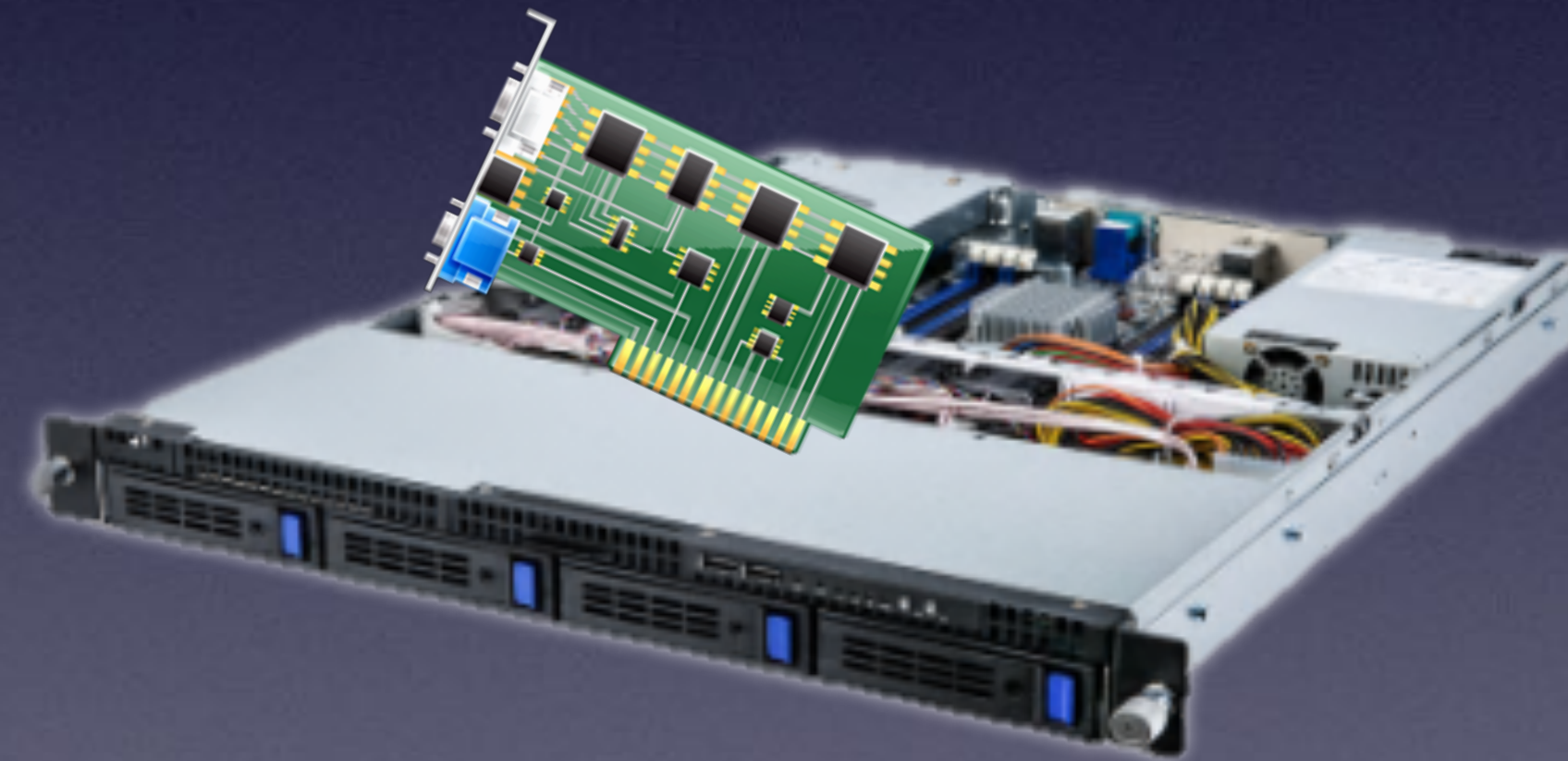
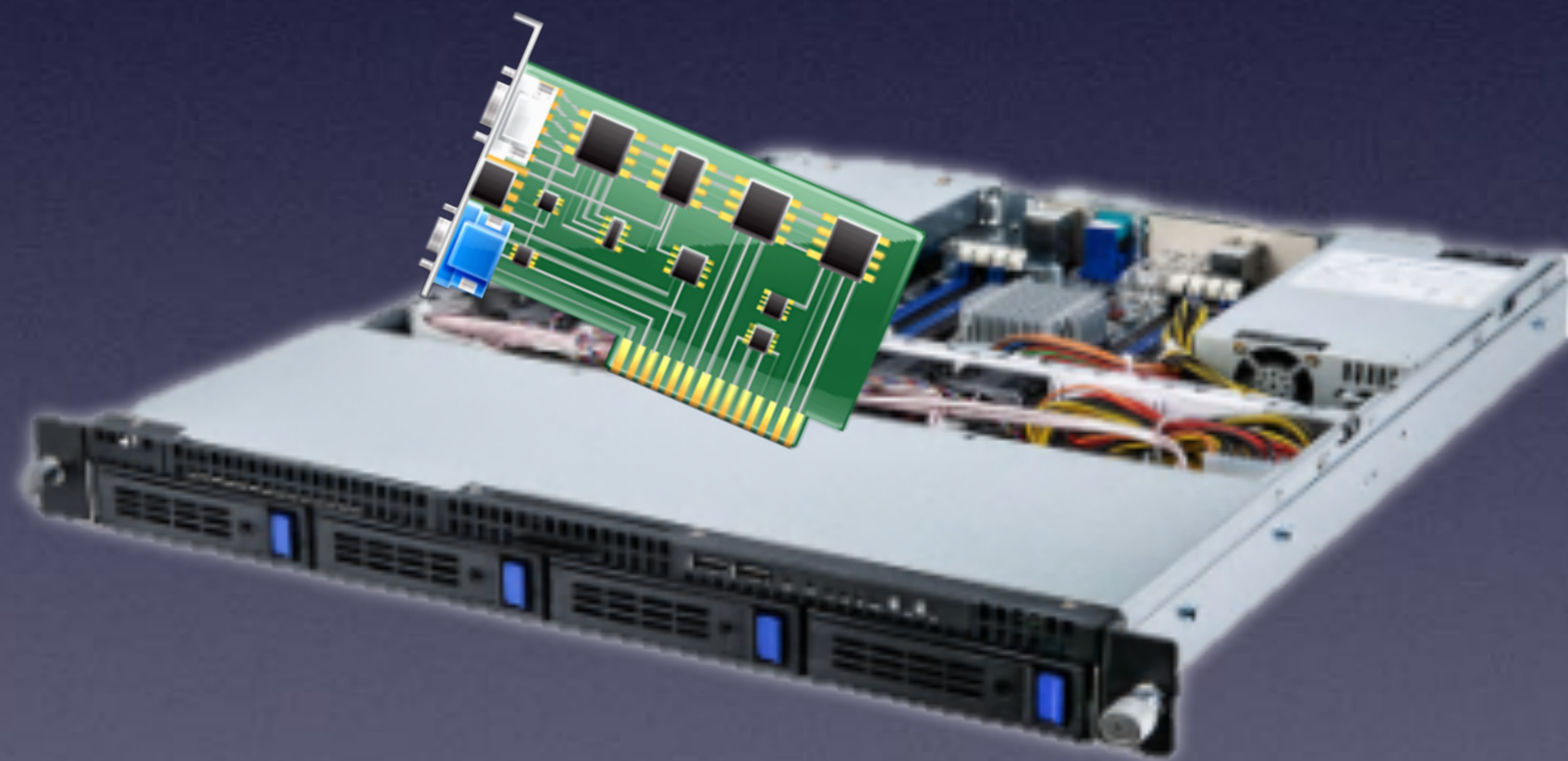


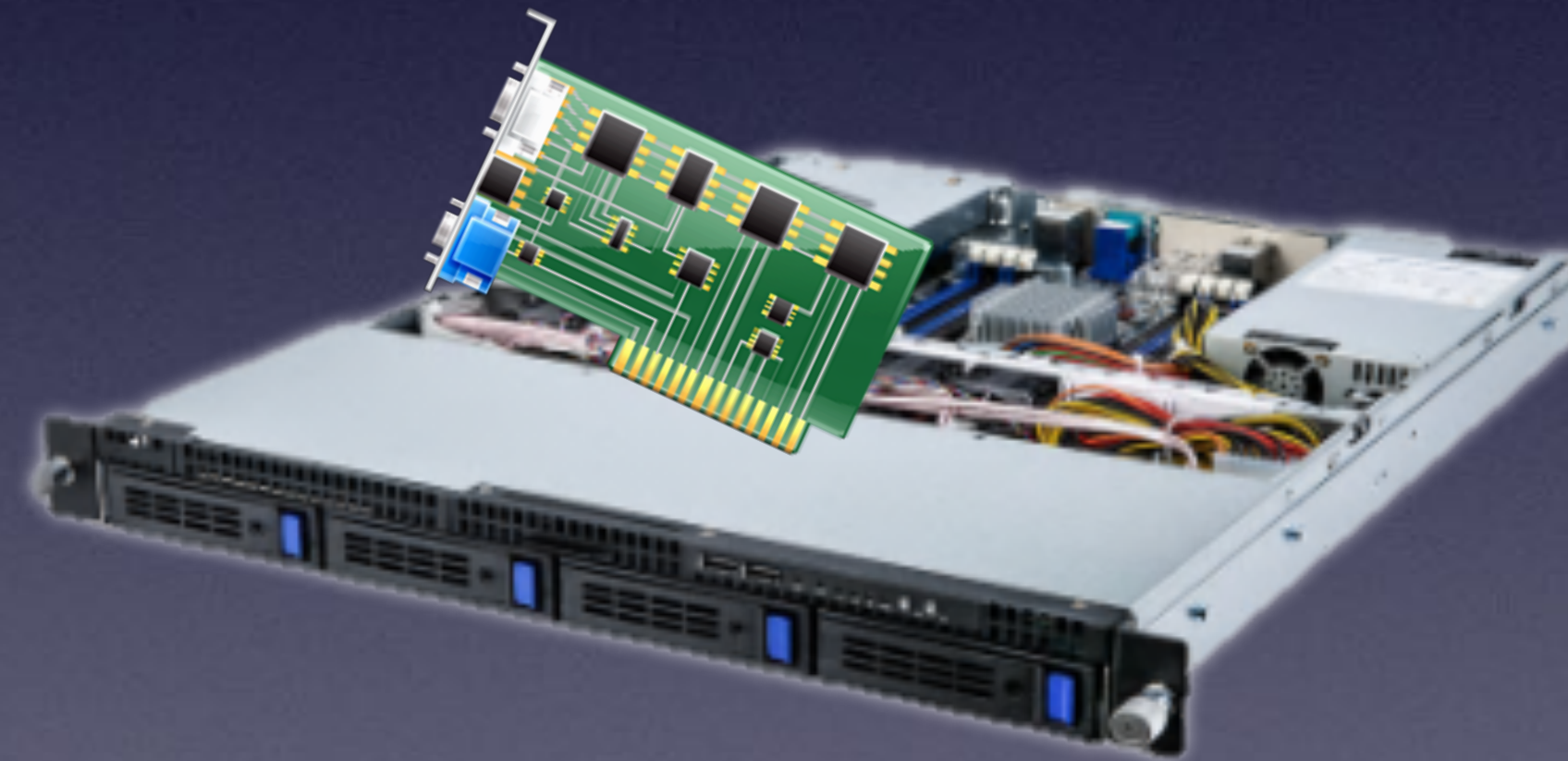
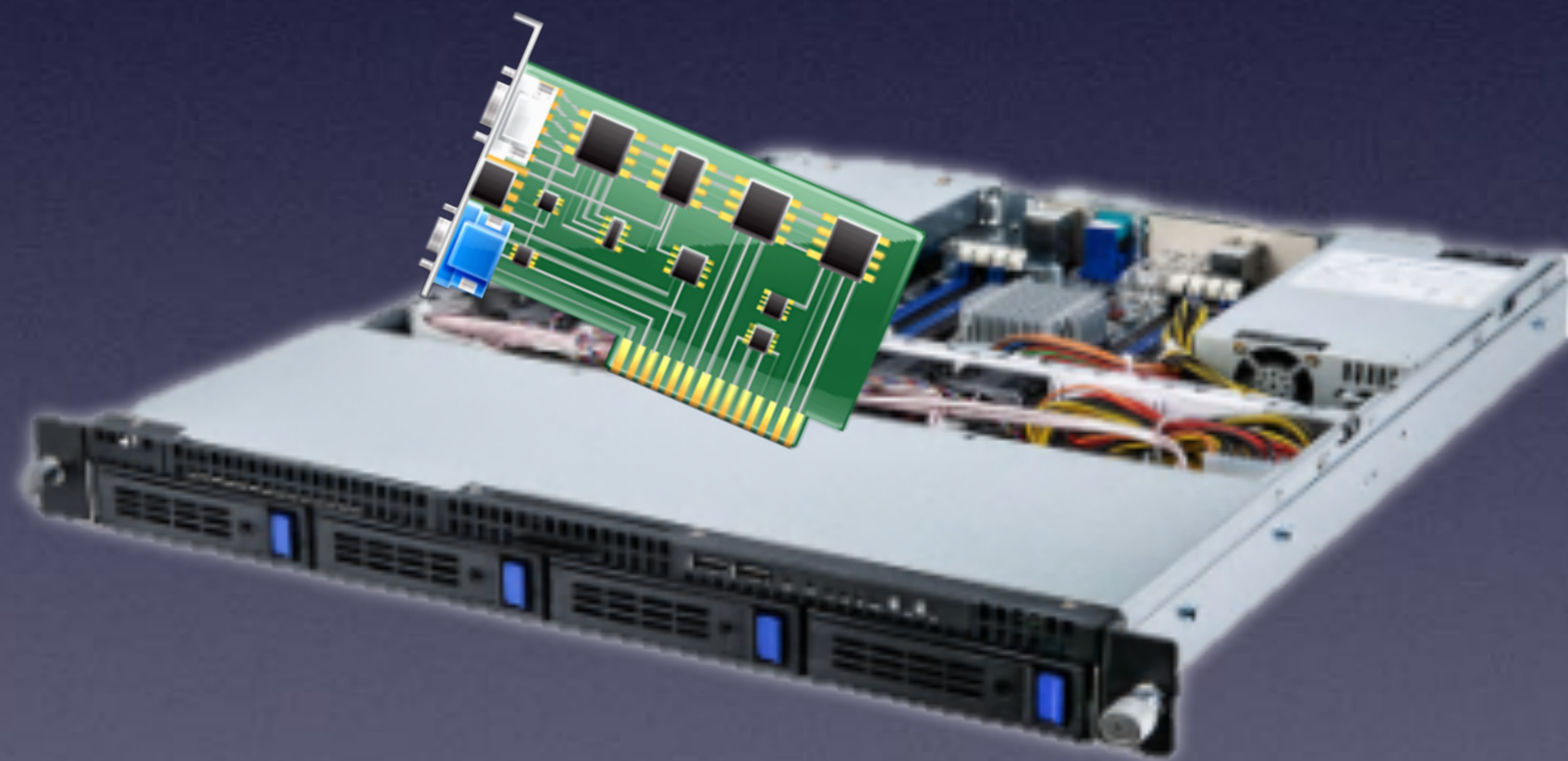
About Me

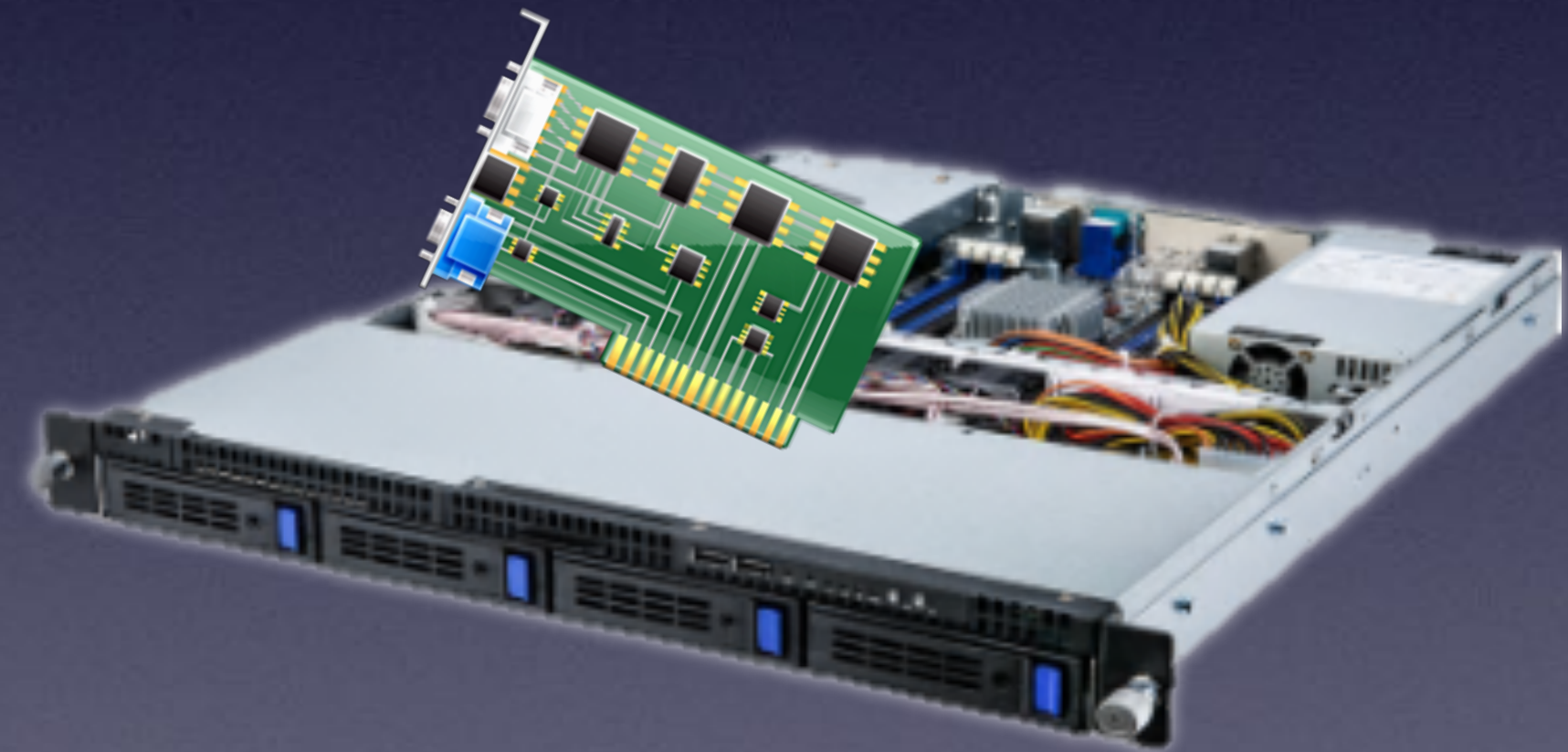
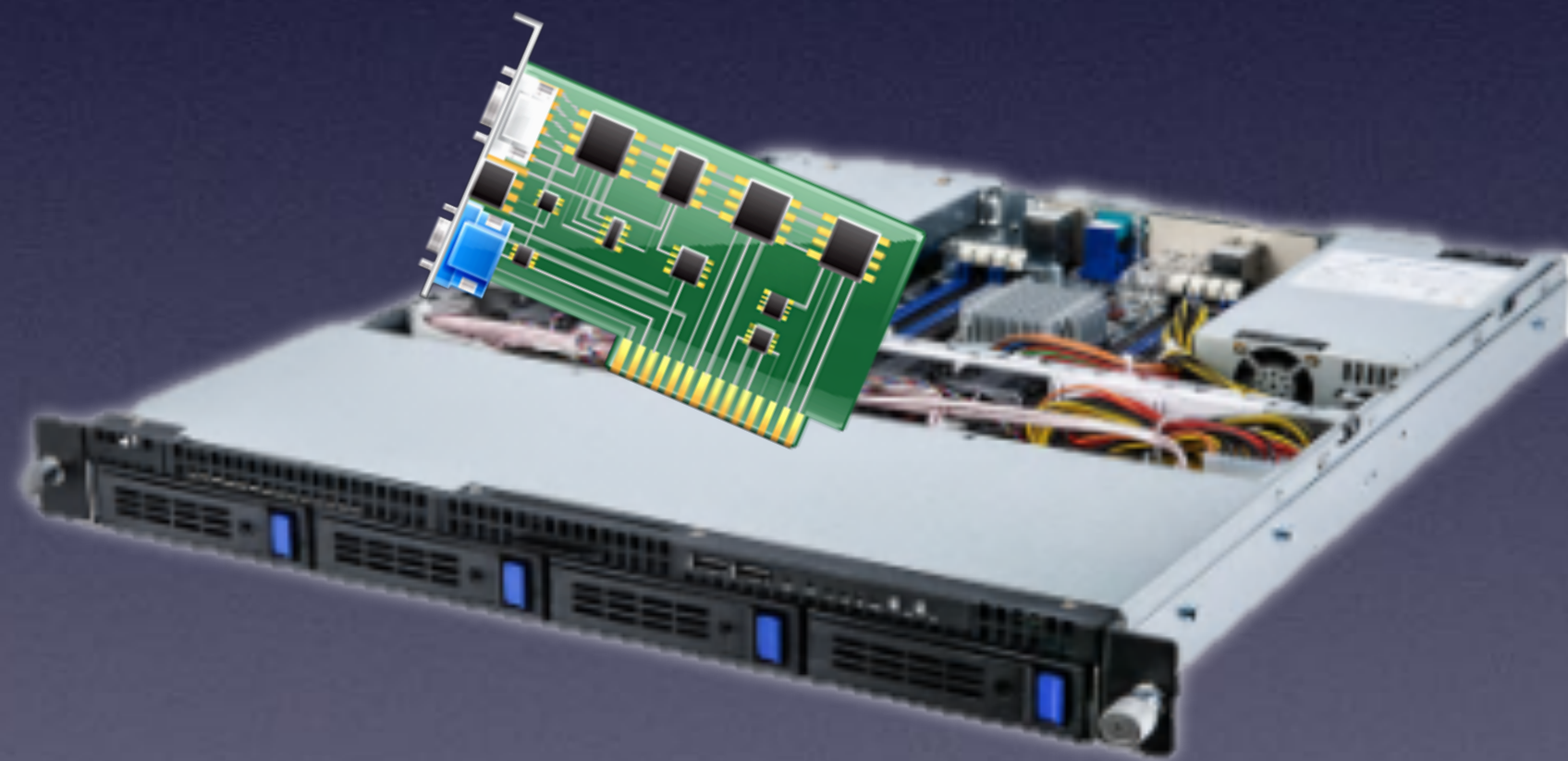
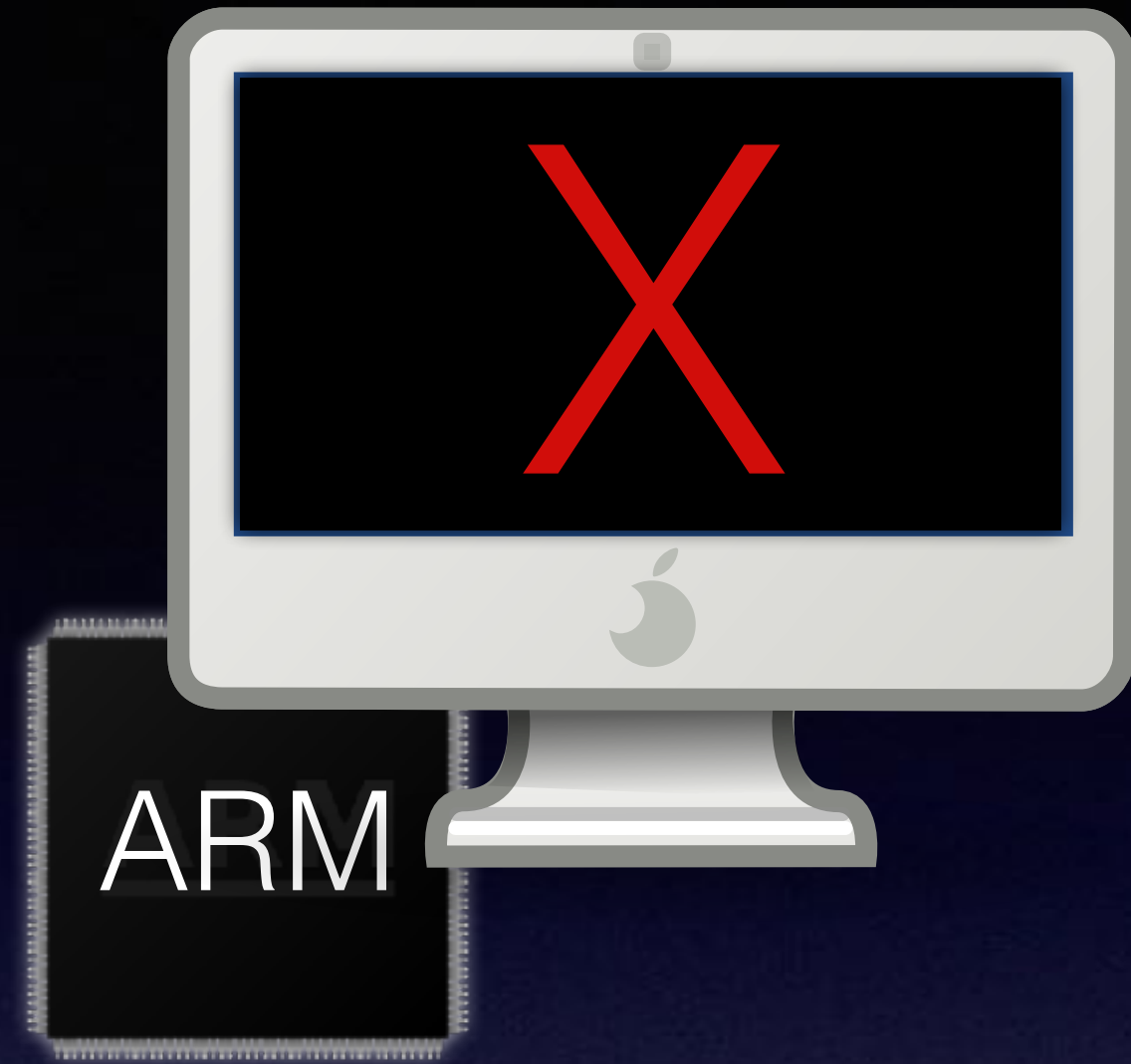
- Alexander Graf
- KVM and QEMU developer for SUSE
 - Server class PowerPC KVM port
 - Nested SVM
- Founding member of SUSE ARM team
- U-Boot UEFI support

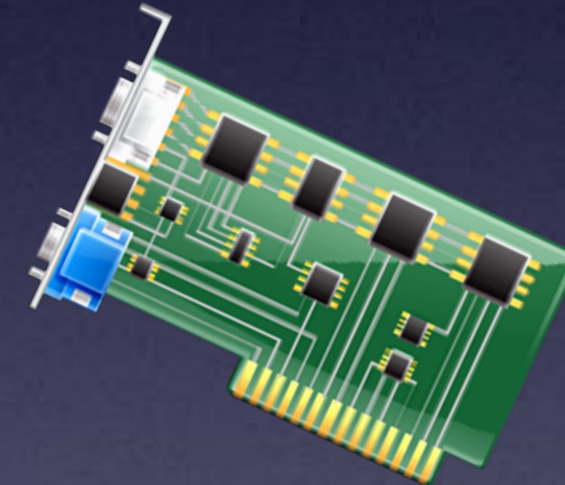
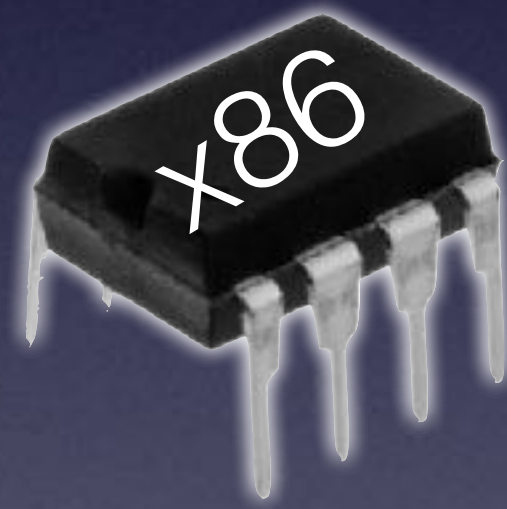
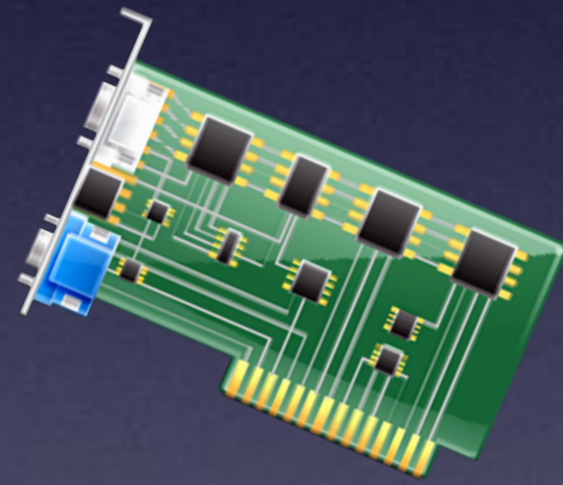
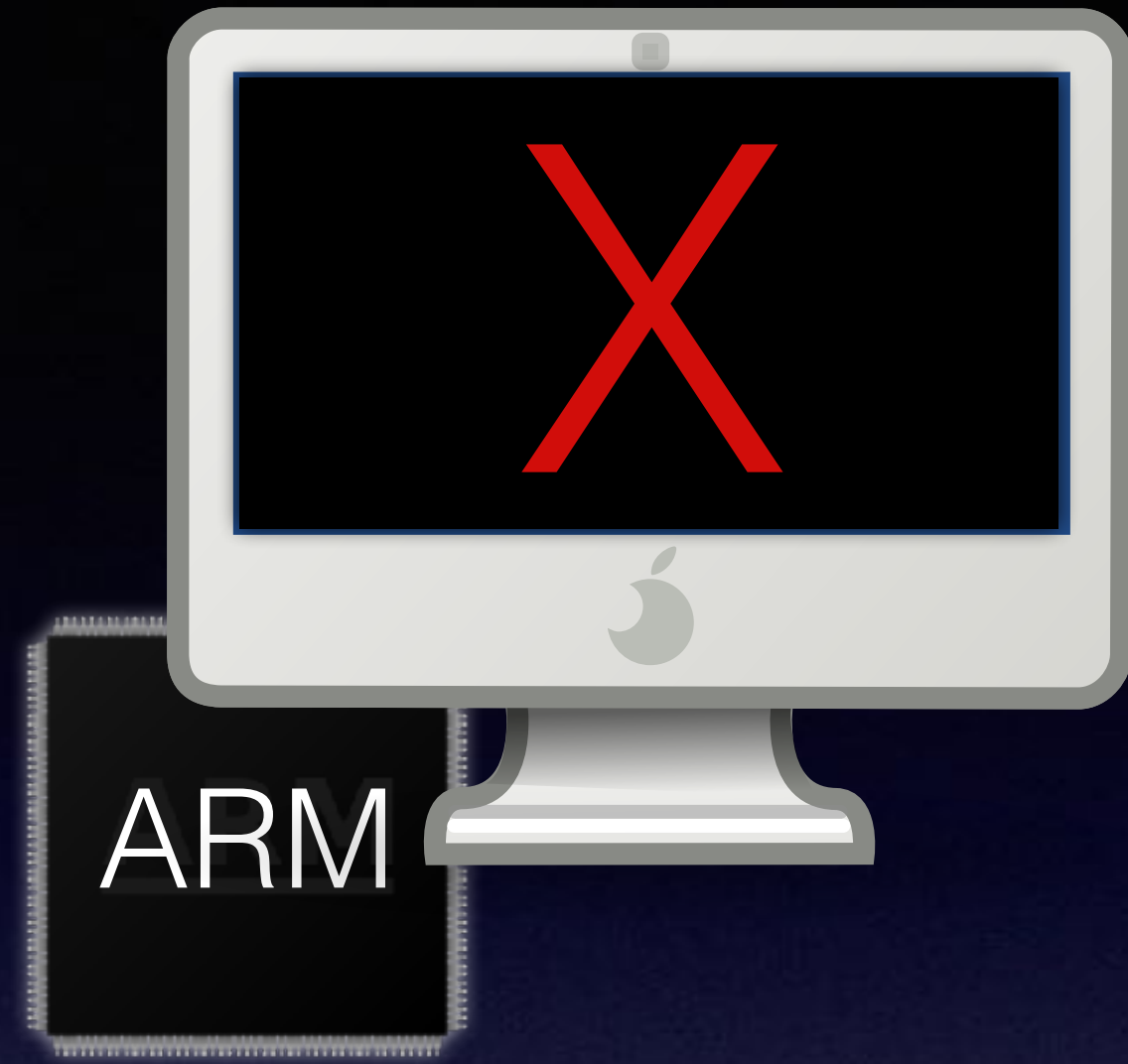










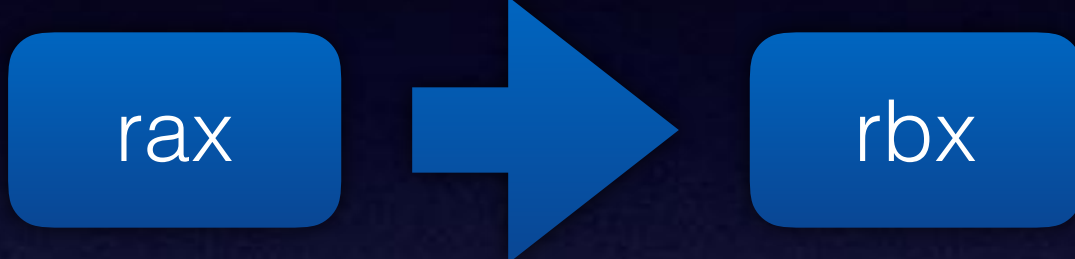


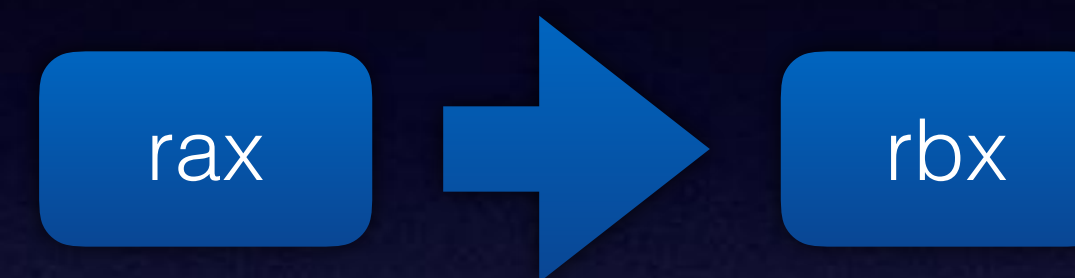
ARM

x86

x0	x1	x2	x3	x4
x5	x6	x7	x8	x9
x10	x11	x12	x13	x14
x15	x16	x17	x18	x19
x20	x21	x22	x23	x24
x25	x26	x27	x28	x29
x30	xzr			

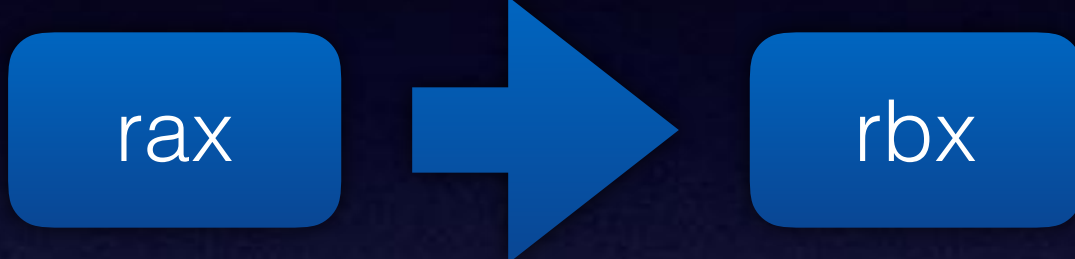
rax	rbx	rcx	rdx	rbp
rsp	rsi	rdi	r8	r9
r10	r11	r12	r13	r14
r15				





aa0003e1 mov x1, x0

48 89 c3 mov %rax,%rbx



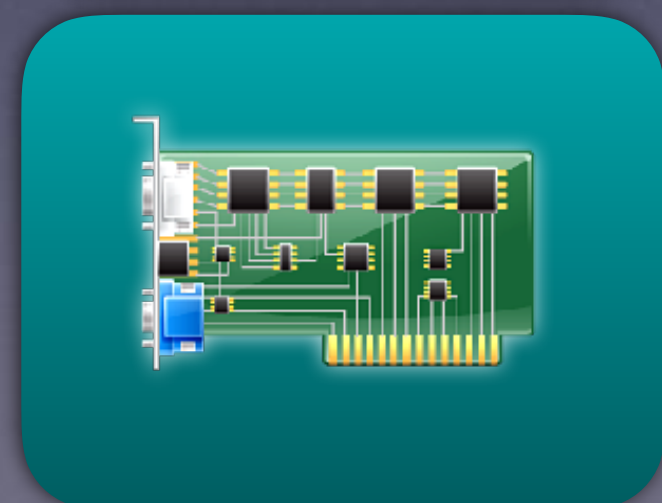
aa0003e1 mov x1, x0

48 89 c3 mov %rax,%rbx



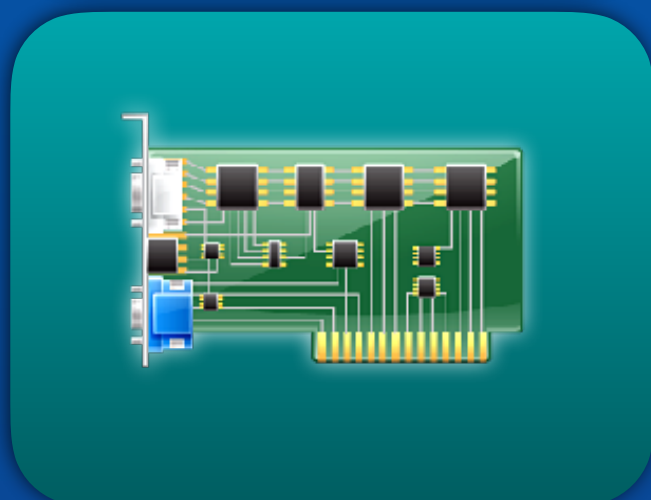


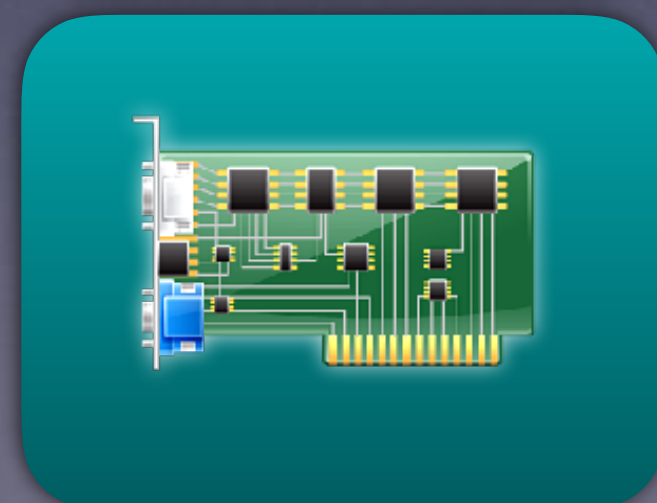


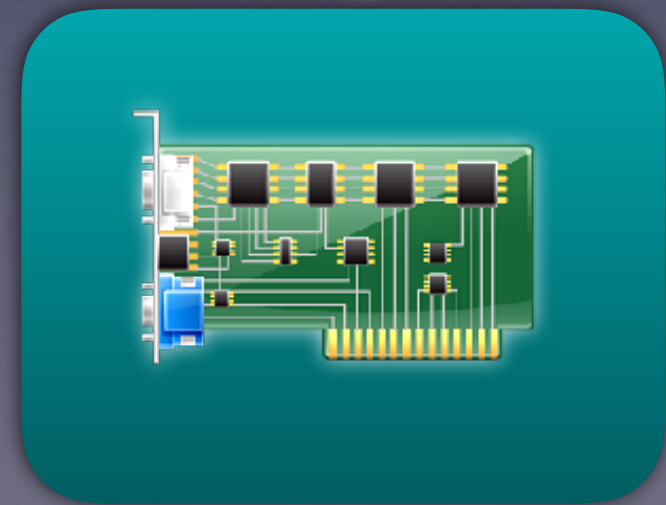




virtual

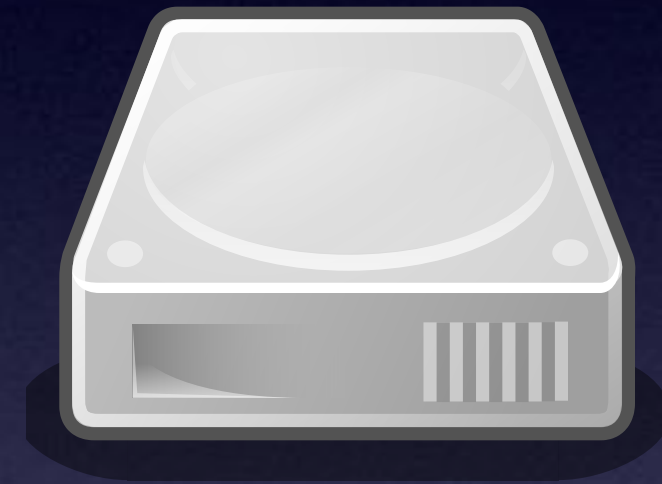


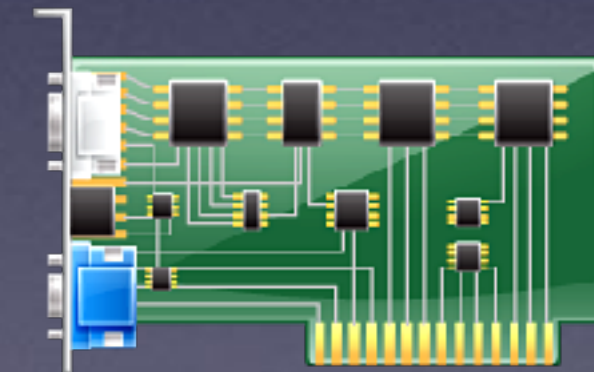
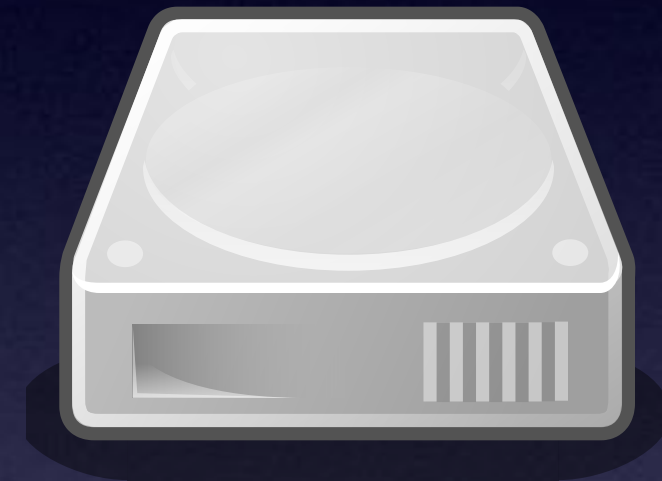


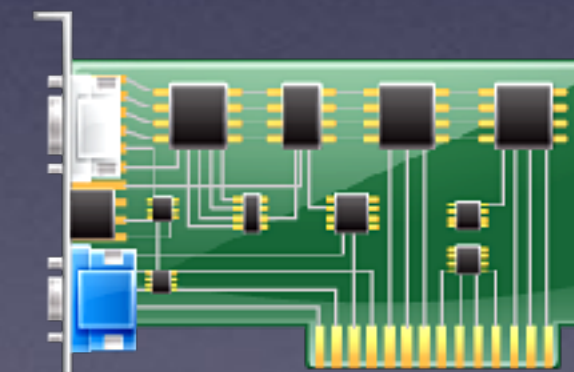
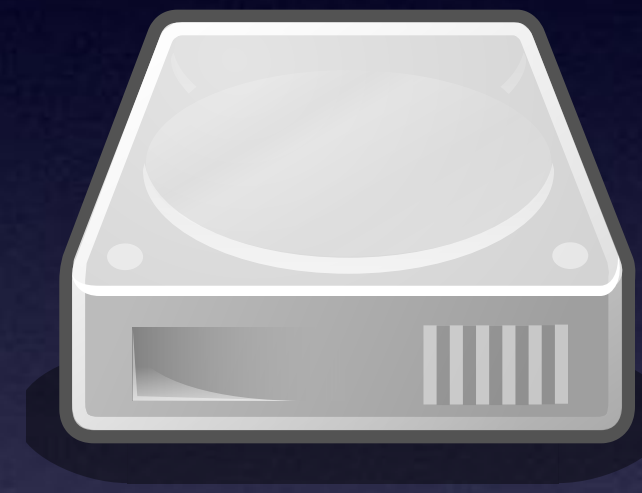


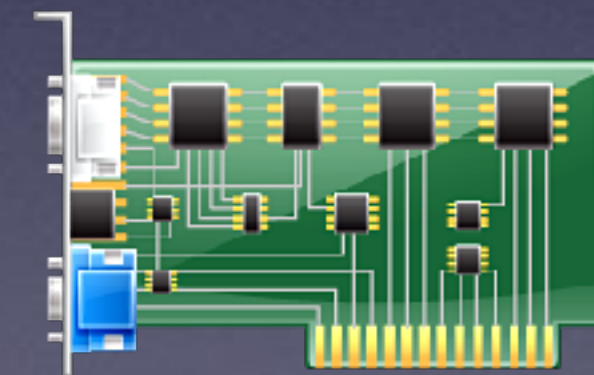
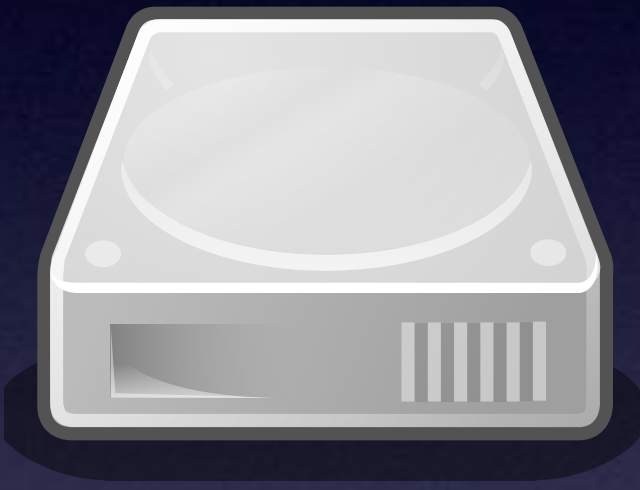










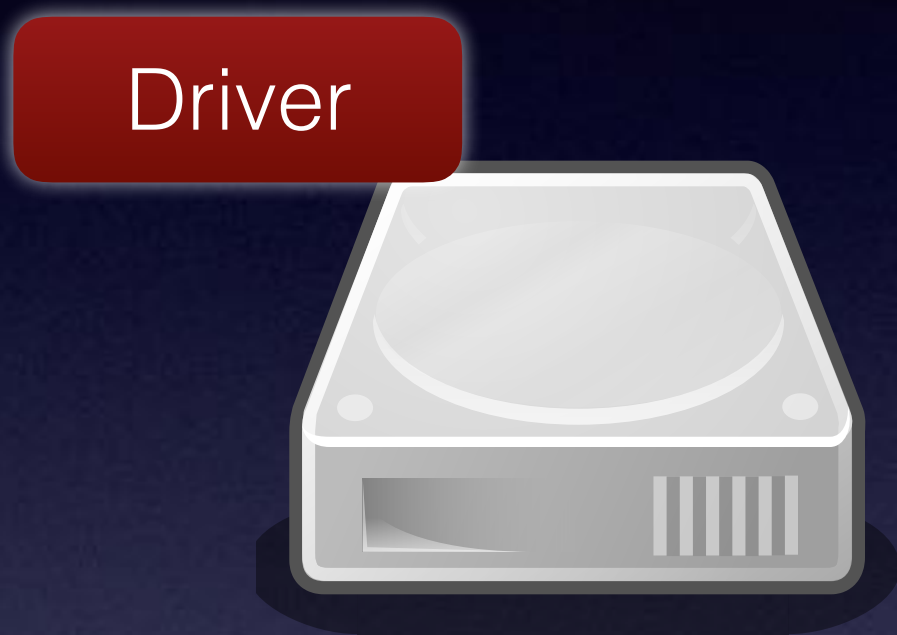




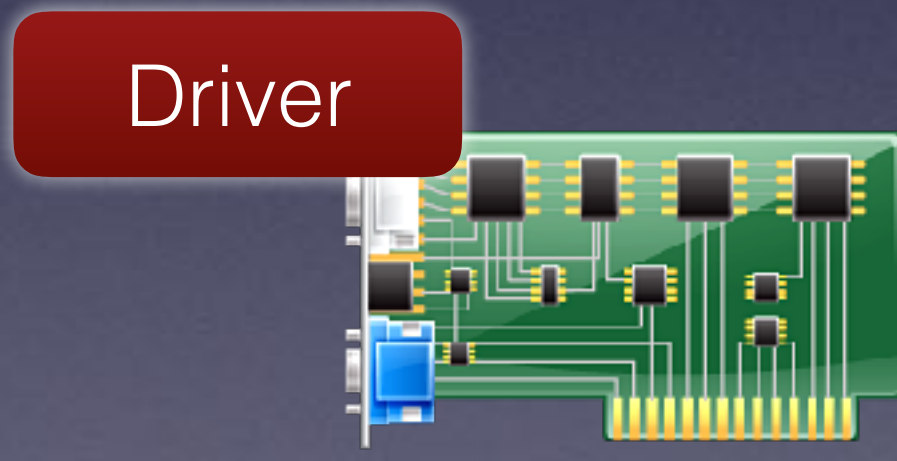
Driver



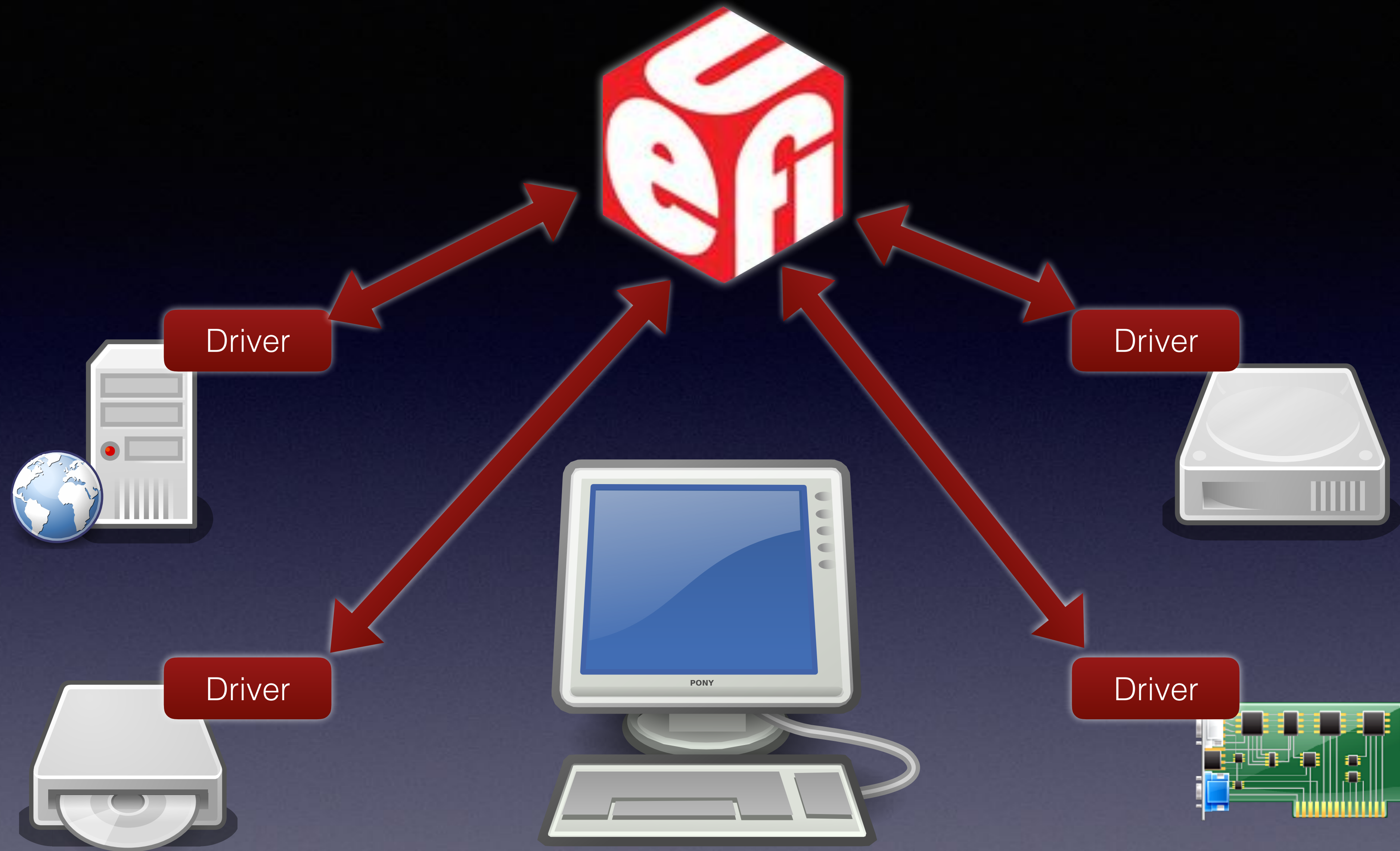
Driver



Driver



Driver



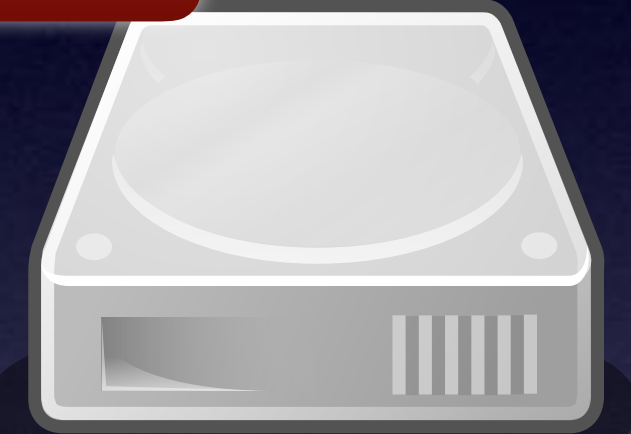


Driver





Driver





Driver









Driver





gBS->InstallMultipleProtocolInterfaces(■)

Driver





gBS->InstallMultipleP

Driver



```
efi_status_t (*supported)(...)  
efi_status_t (*start)(...)  
efi_status_t (*stop)(...)  
-----  
u32 version  
void *image_handle  
void *driver_handle
```




Driver

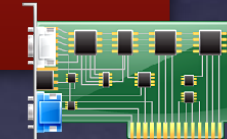


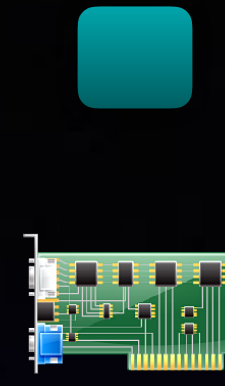


Driver



PCI driver





Driver

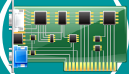


PCI driver



Driver

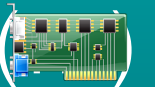


```
efi_status_t (*supported)(  
efi_status_t (*start)(...  
efi_status_t (*stop)(...  
-----  
u32 version  
void *image_handle  
void *driver_handle
```




Driver

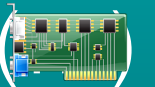


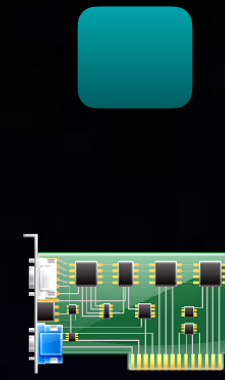
```
efi_status_t (supported)(...)  
efi_status_t (*start)()  
efi_status_t (*stop)(...)  
-----  
u32 version  
void *image_handle  
void *driver_handle
```



Driver



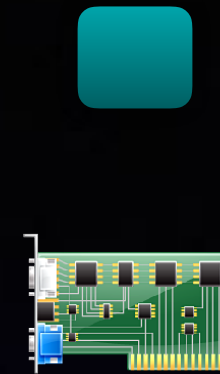
```
efi_status_t (*supported)(...)  
efi_status_t (*start)()  
efi_status_t (*stop)(...)  
-----  
u32 version  
void *image_handle  
void *driver_handle
```

gBS->InstallMultipleProtocolInterfaces(■)

Driver



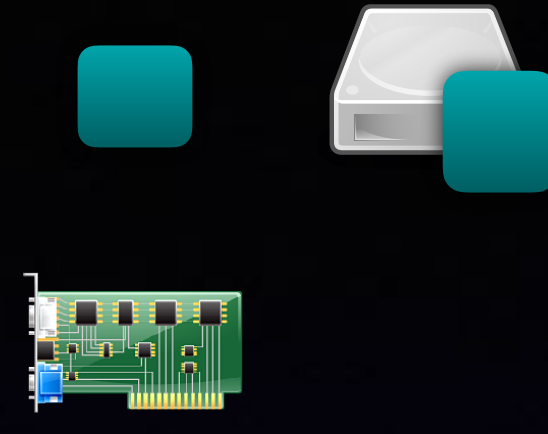


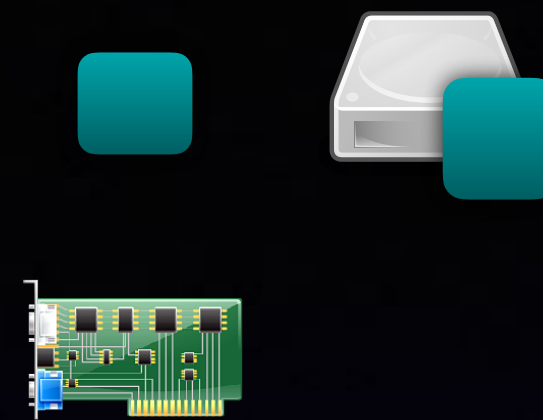
gBS->InstallMultipleProtocolInterfaces()

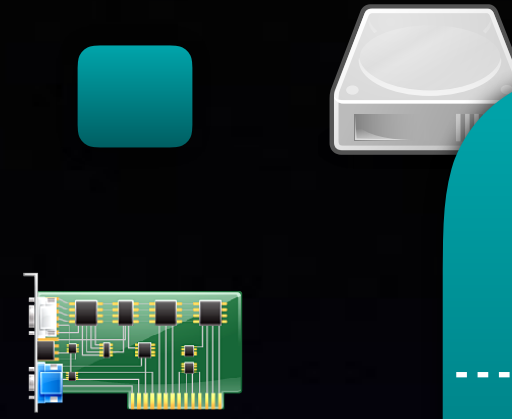
Driver



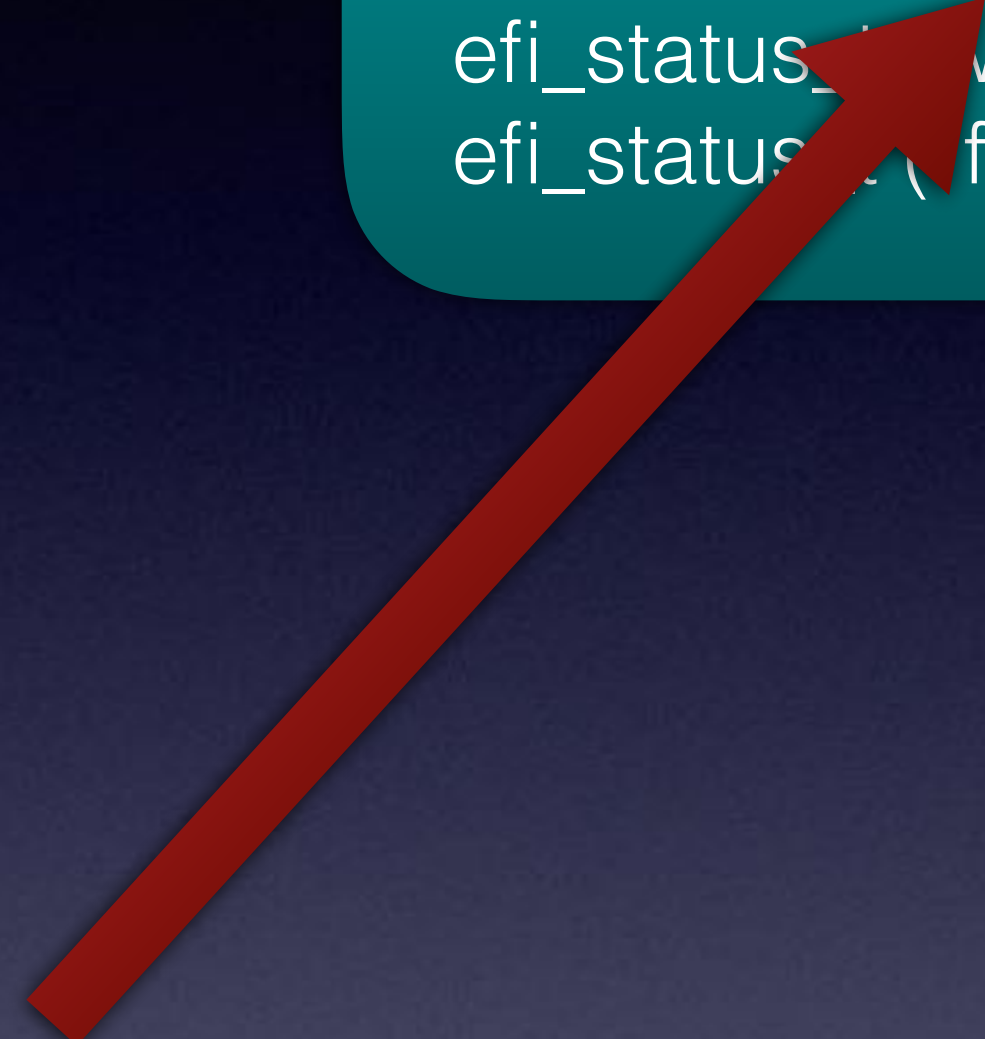
```
u64 revision  
struct block_io_media *m  
-----  
efi_status_t (*reset)(...)  
efi_status_t (*read)(...)  
efi_status_t (*write)(...)  
efi_status_t (*flush)(...)
```

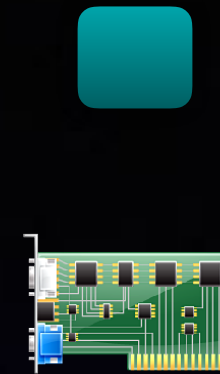







```
u64 revision
struct block_io_media *m
-----
efi_status_t (*reset)(...)
efi_status_t (*read)(...)
efi_status_t (*write)(...)
efi_status_t (*flush)(...)
```





virtual



```
u64 revision
struct block_io_media *m
-----
efi_status_t (*reset)(...)
efi_status_t (*read)(...)
efi_status_t (*write)(...)
efi_status_t (*flush)(...)
```

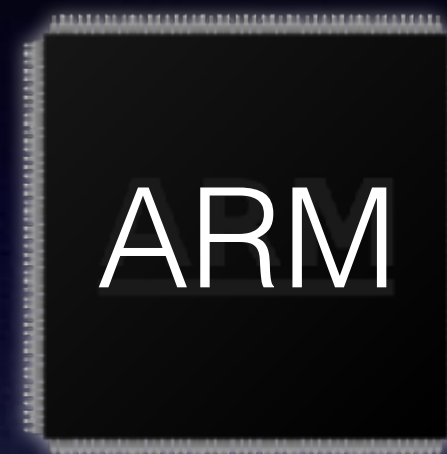



virtual



```
u64 revision
struct block_io_media *m
-----
efi_status_t (*reset)(...)
efi_status_t (*read)(...)
efi_status_t (*write)(...)
efi_status_t (*flush)(...)
```

```
0x00:  01 00 00 00 00 00 00 00
0x08:  f0 ee dd cc bb aa 99 88
0x10:  00 ee dd cc bb aa 99 88
0x18:  00 ed dd cc bb aa 99 88
0x20:  00 ec dd cc bb aa 99 88
0x28:  00 eb dd cc bb aa 99 88
```



virtual



```
u64 revision
struct block_io_media *m
-----
efi_status_t (*reset)(...)
efi_status_t (*read)(...)
efi_status_t (*write)(...)
efi_status_t (*flush)(...)
```

0x00:	01	00	00	00	00	00	00	00
0x08:	f0	ee	dd	cc	bb	aa	99	88
0x10:	00	ee	dd	cc	bb	aa	99	88
0x18:	00	ed	dd	cc	bb	aa	99	88
0x20:	00	ec	dd	cc	bb	aa	99	88
0x28:	00	eb	dd	cc	bb	aa	99	88



u32 id

efi_status_t (*reset)(...)

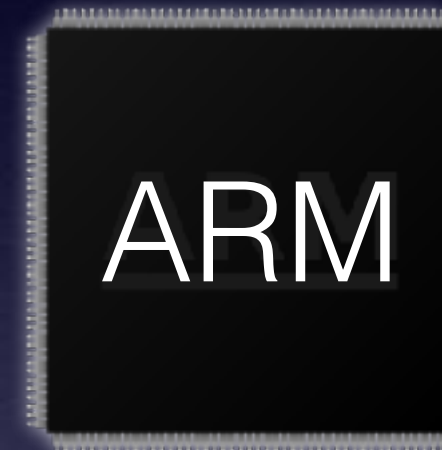
0x00:	01	00	00	00	XX	XX	XX	XX
0x08:	f0	ee	dd	cc	bb	aa	99	88



u32 id

efi_status_t (*reset)(...)

0x00:	01 00 00 00	XX XX XX XX
0x08:	f0 ee dd cc	bb aa 99 88



u32 id

efi_status_t (*reset)(...)

0x00:	01	00	00	00	XX	XX	XX	XX
0x08:	f0	ee	dd	cc	bb	aa	99	88



virtual



```
u64 revision
struct block_io_media *m
-----
efi_status_t (*reset)(...)
efi_status_t (*read)(...)
efi_status_t (*write)(...)
efi_status_t (*flush)(...)
```




virtual

x86

```
u64 revision
struct block_io_media *m
-----
efi_status_t (*reset)(...)
efi_status_t (*read)(...)
efi_status_t (*write)(...)
efi_status_t (*flush)(...)
```



GRUB

OS



ARM

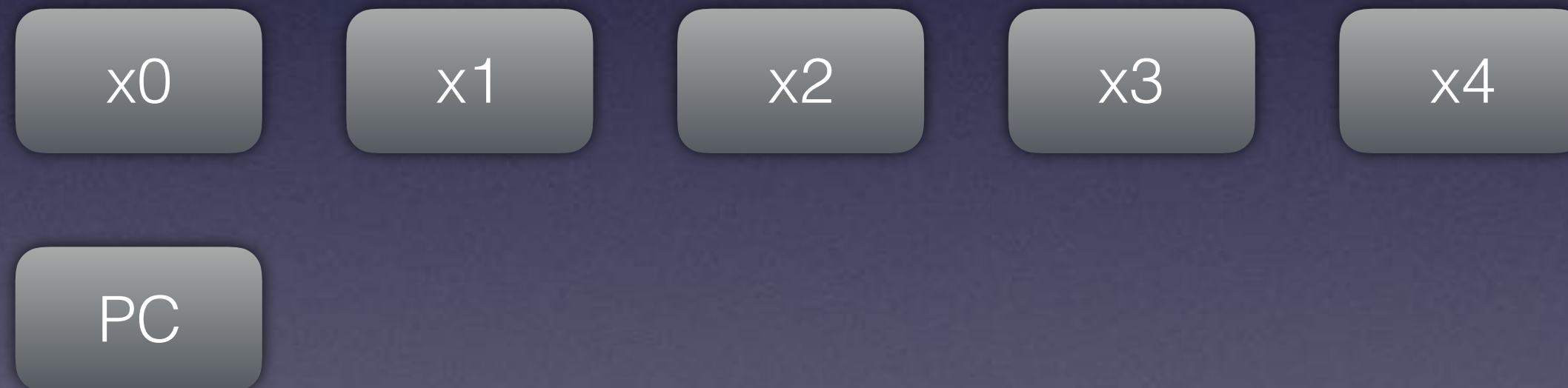
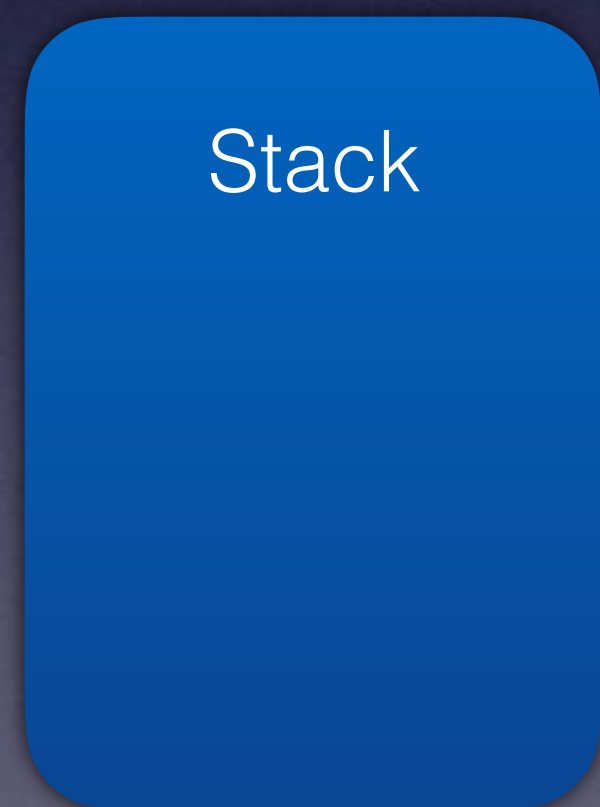




```
efi_status_t (*read)(struct block_io *this, u32 media_id, u64 lba, ulong buffer_size, void*buffer)
```

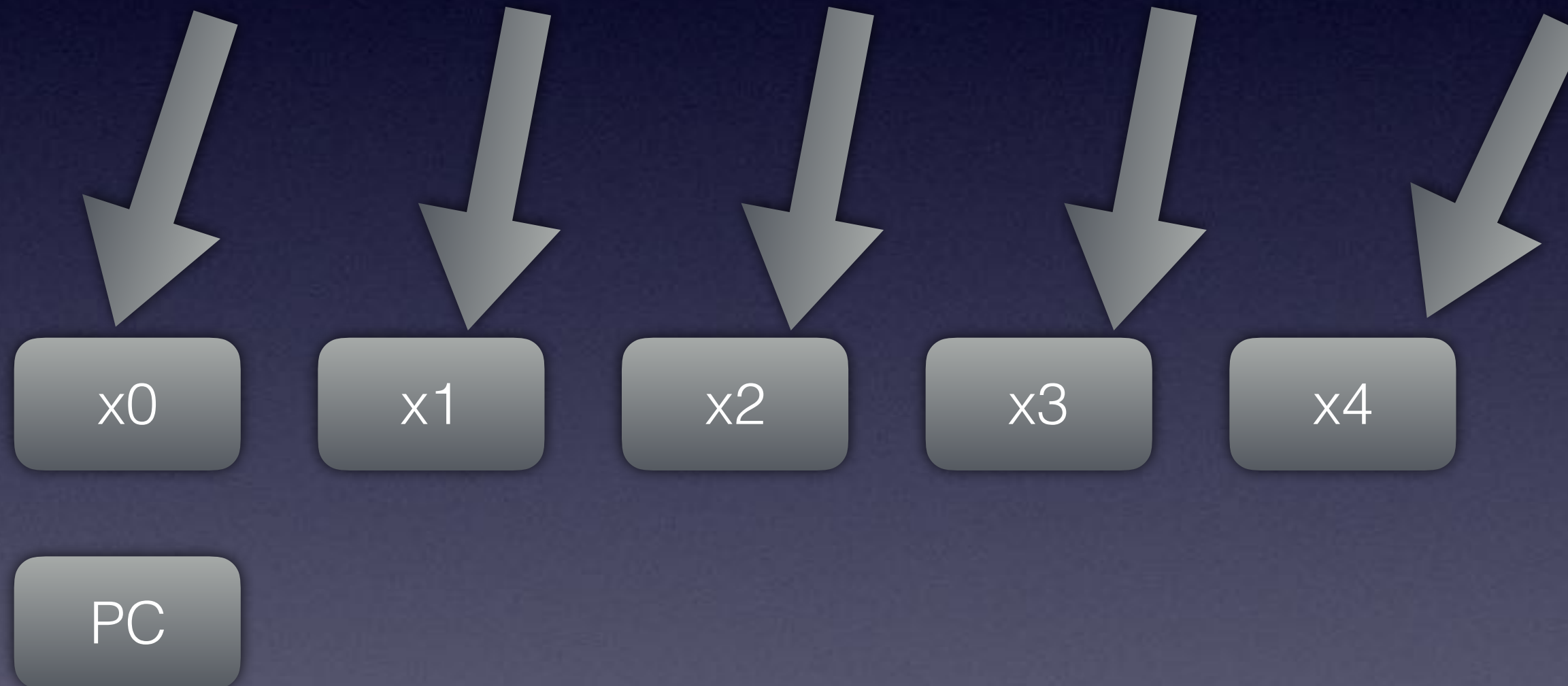
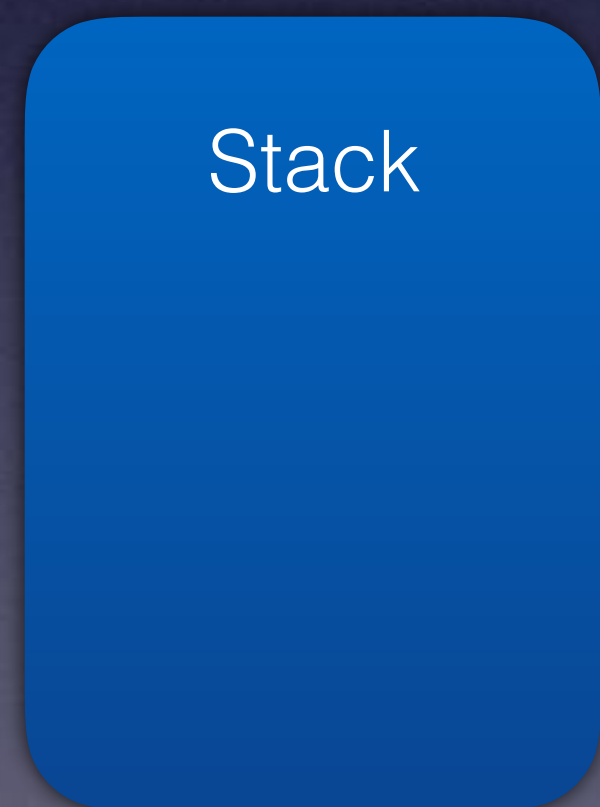



```
efi_status_t (*read)(struct block_io *this, u32 media_id, u64 lba, ulong buffer_size, void*buffer)
```



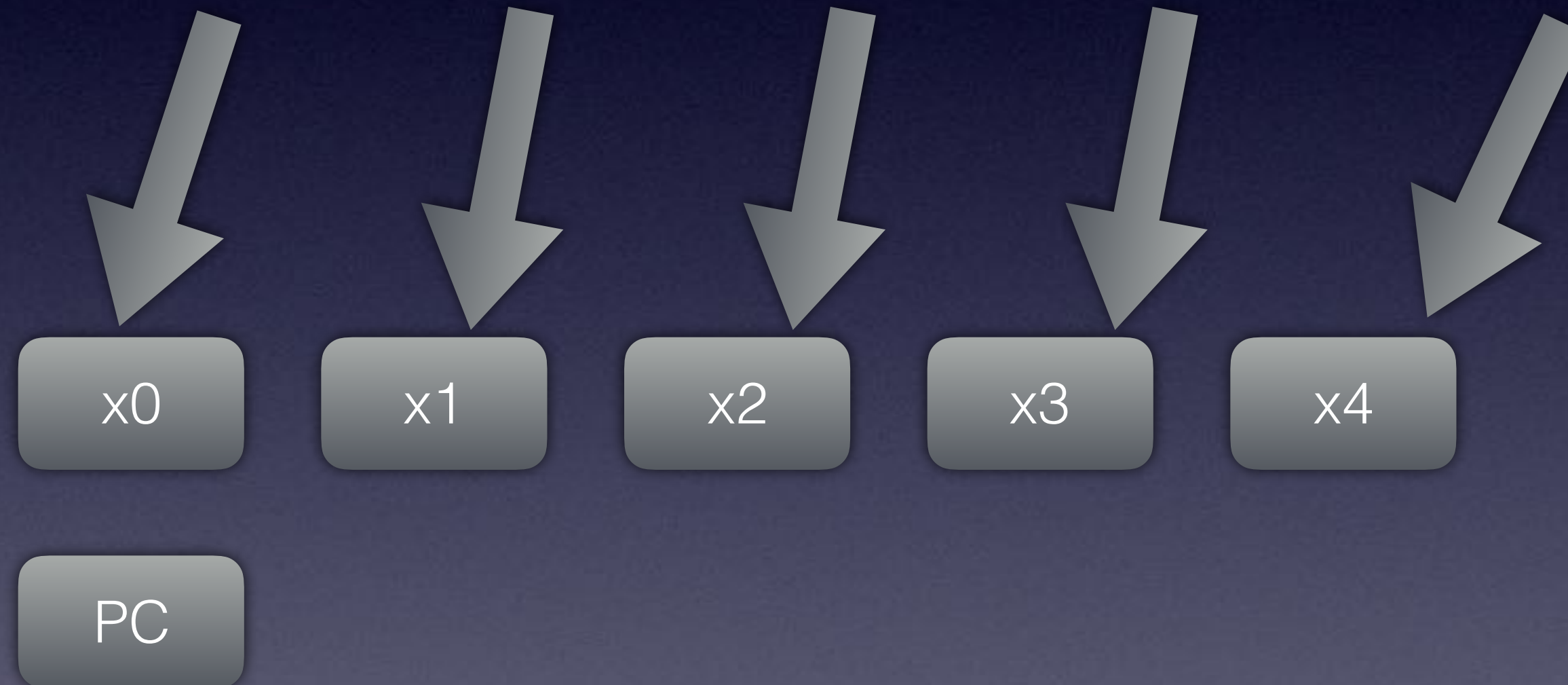
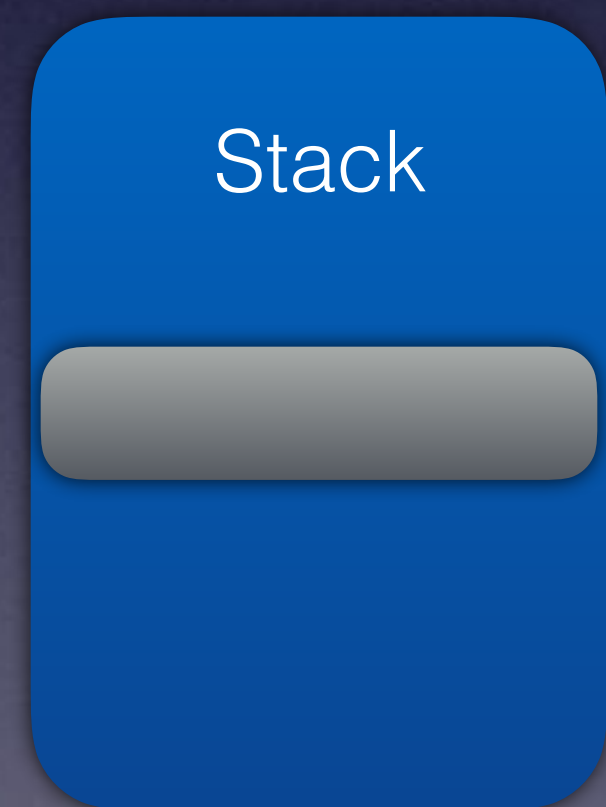


```
efi_status_t (*read)(struct block_io *this, u32 media_id, u64 lba, ulong buffer_size, void*buffer)
```



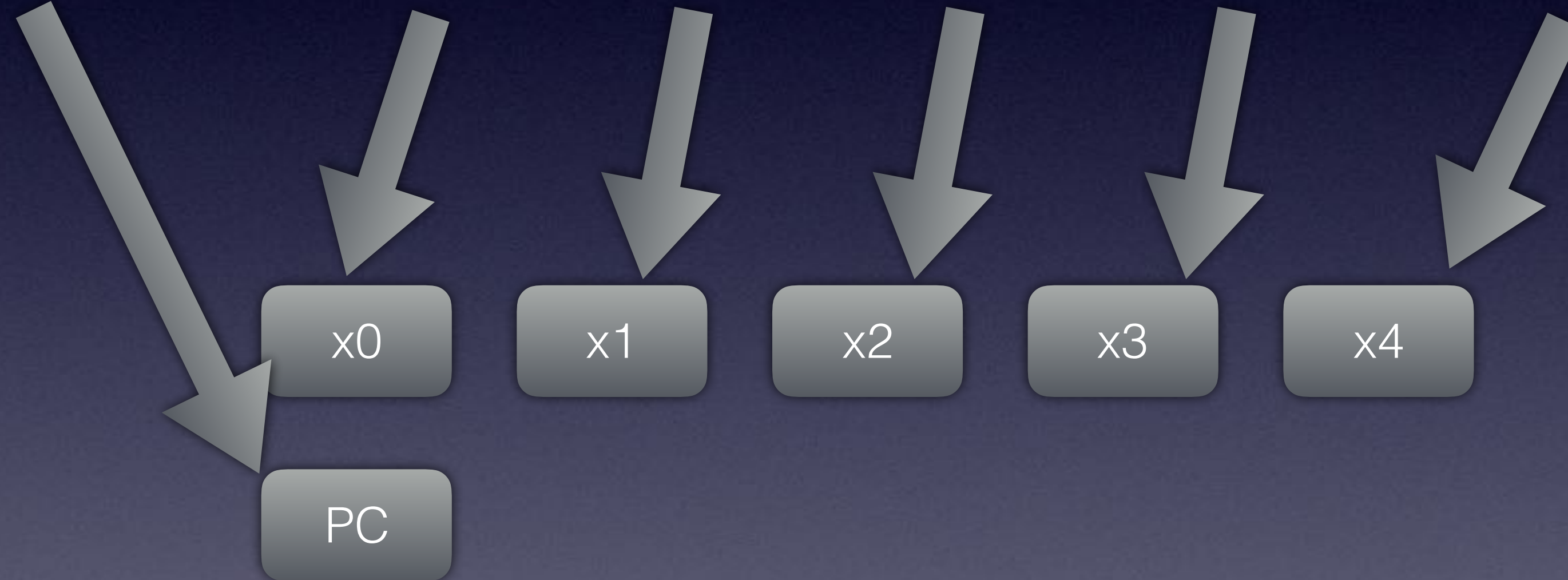
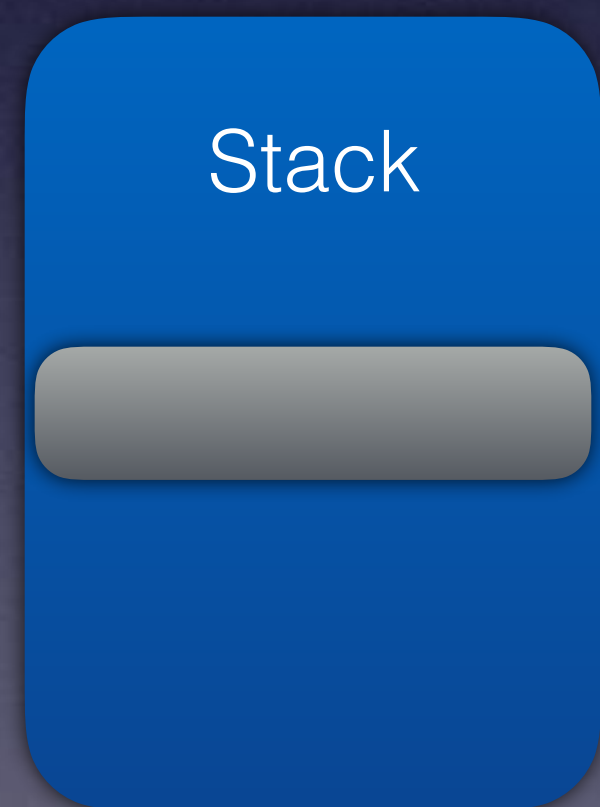


```
efi_status_t (*read)(struct block_io *this, u32 media_id, u64 lba, ulong buffer_size, void*buffer)
```



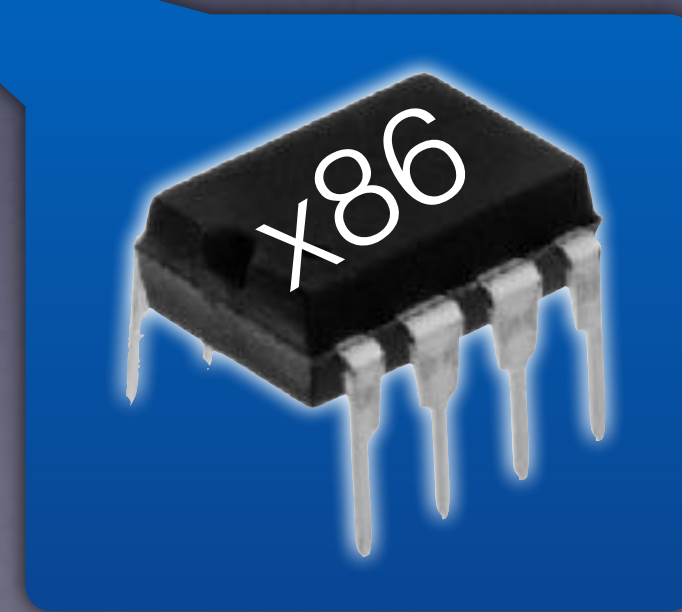
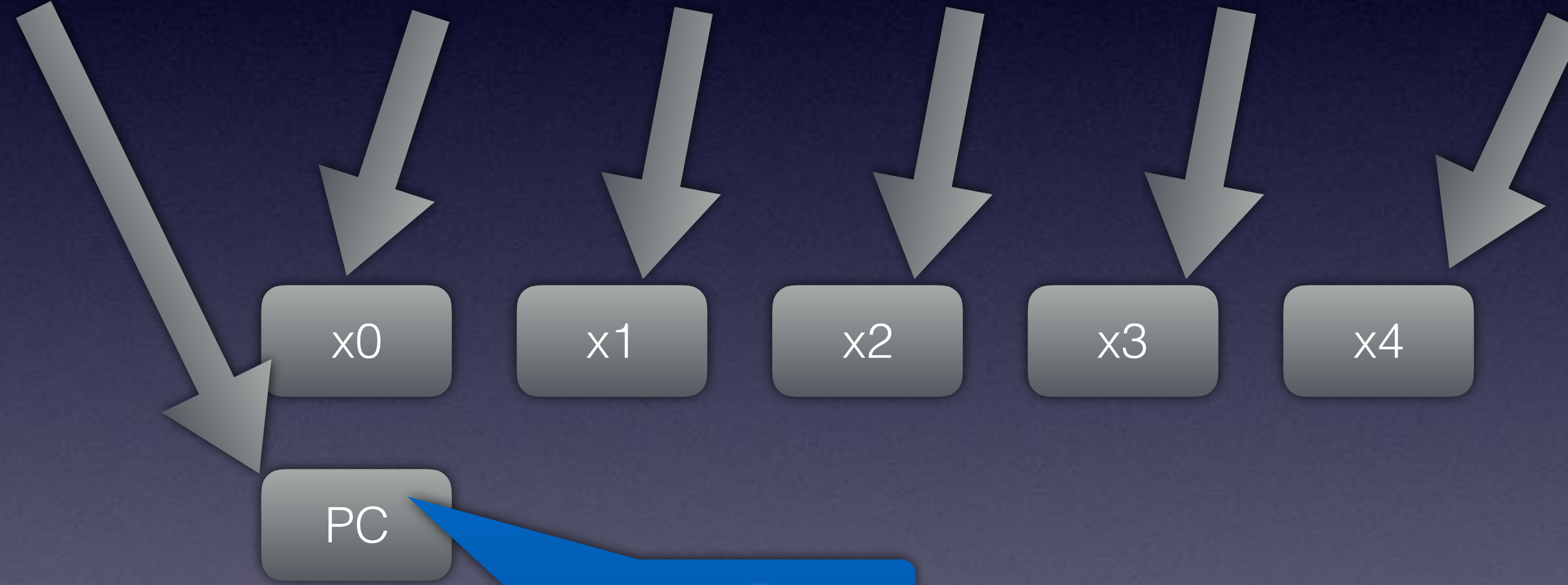
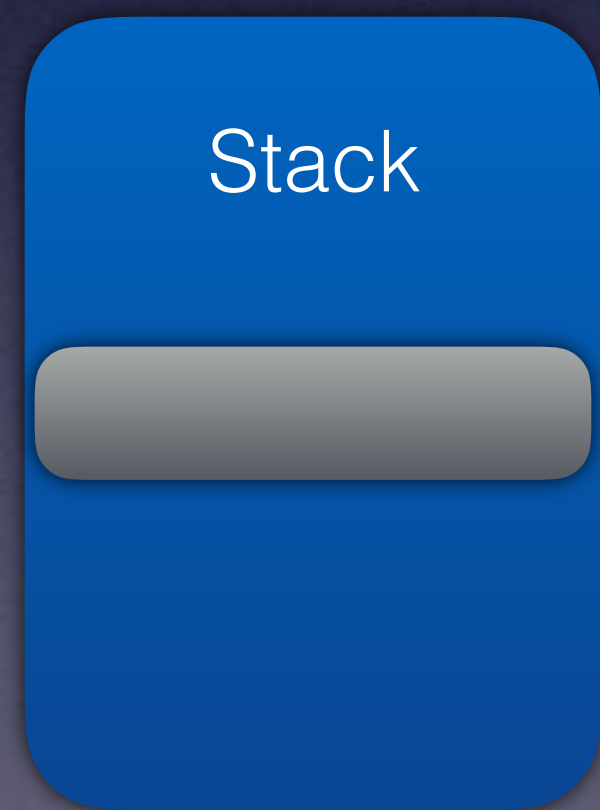


```
efi_status_t (*read)(struct block_io *this, u32 media_id, u64 lba, ulong buffer_size, void*buffer)
```

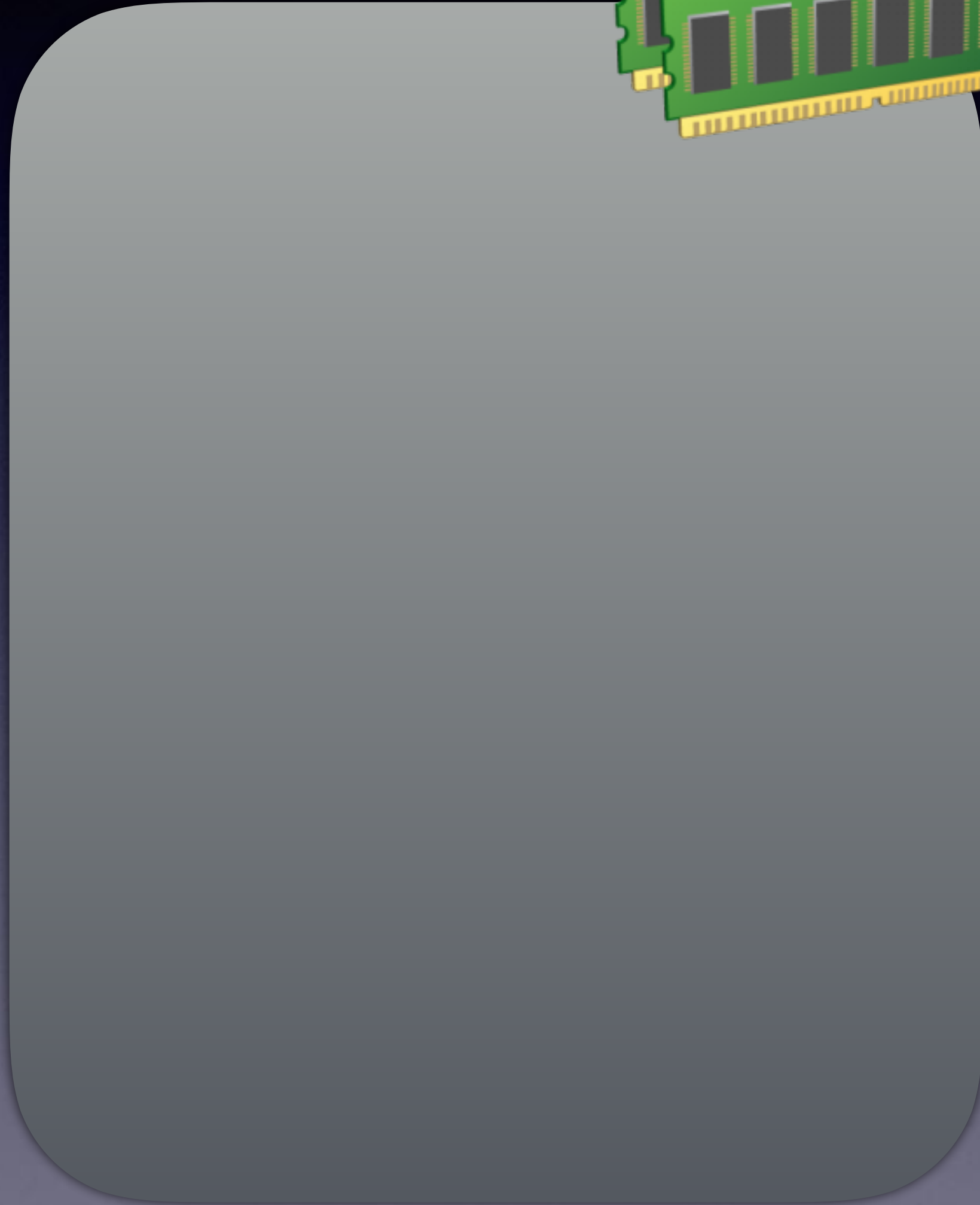
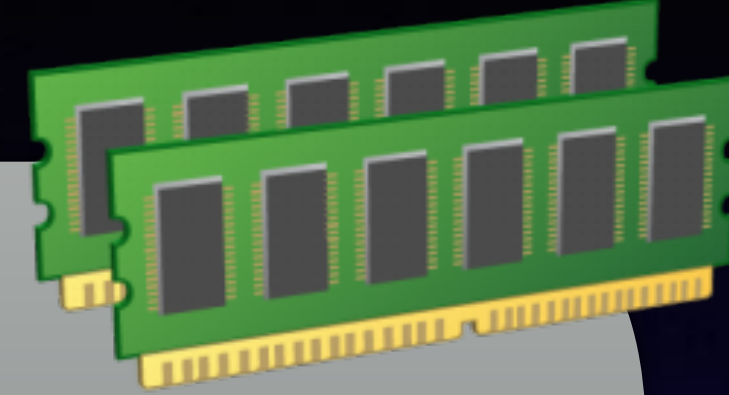




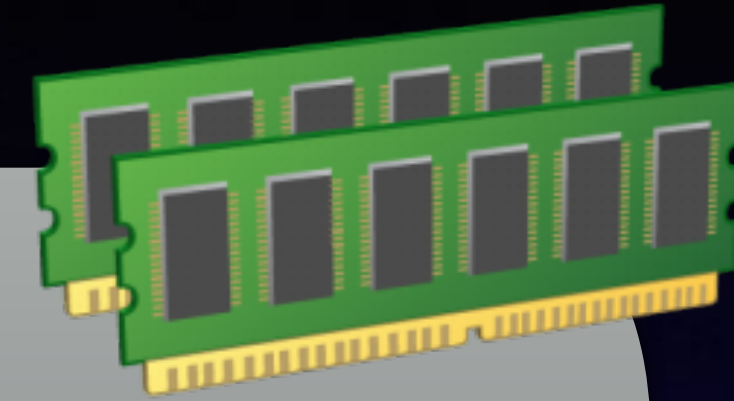
```
efi_status_t (*read)(struct block_io *this, u32 media_id, u64 lba, ulong buffer_size, void*buffer)
```



NX



NX



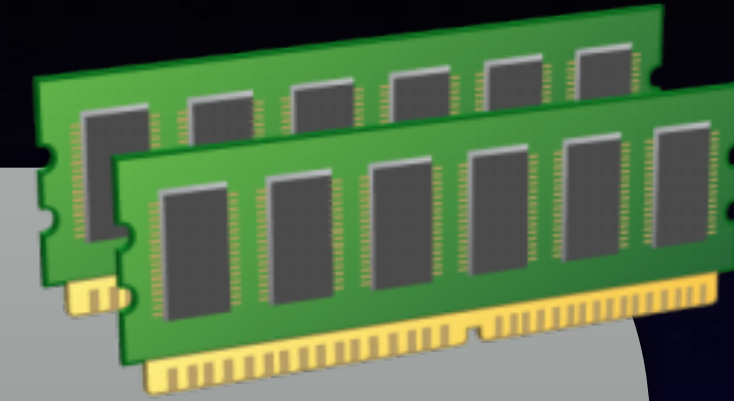
read, write, execute

read, write, execute

read, write, execute

read, write, execute

NX

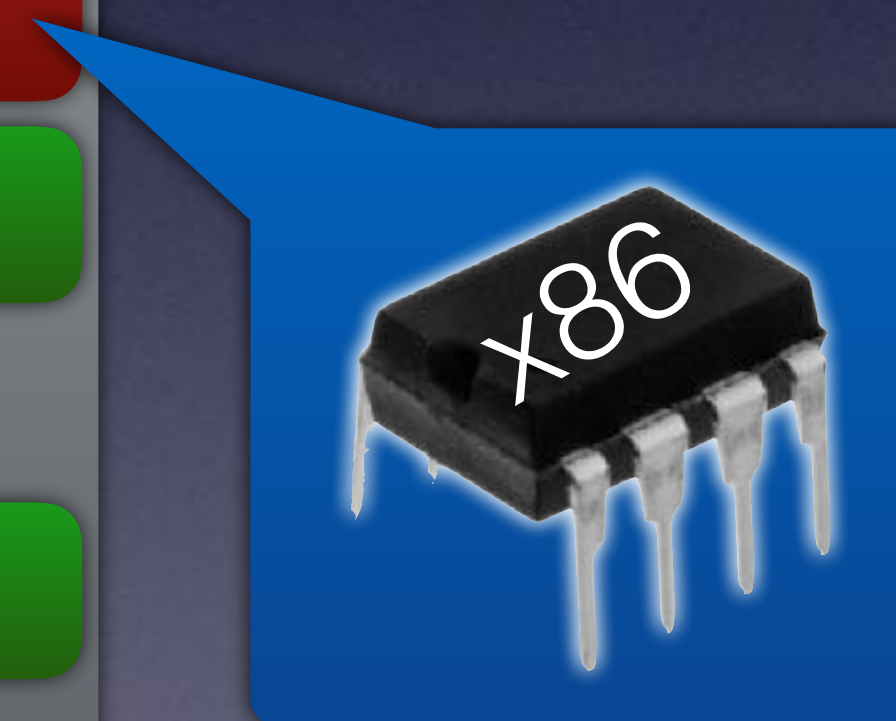


read, write, execute

read, write

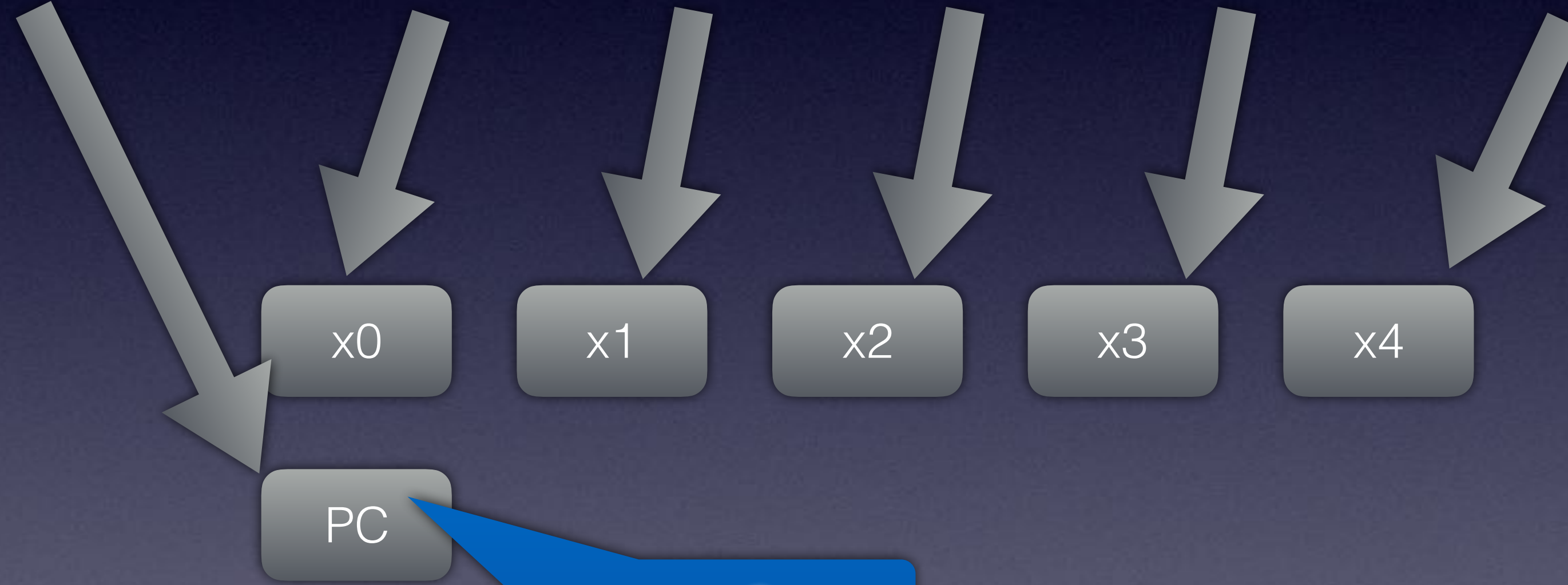
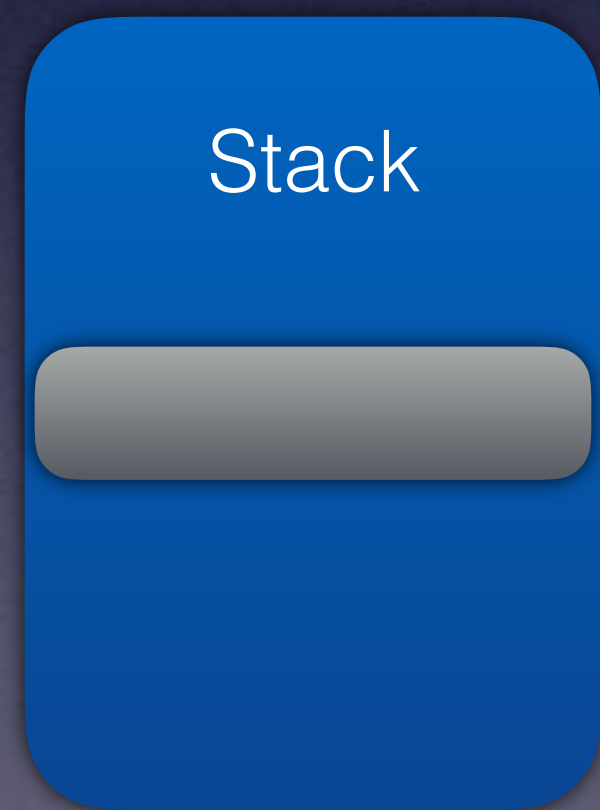
read, write, execute

read, write, execute



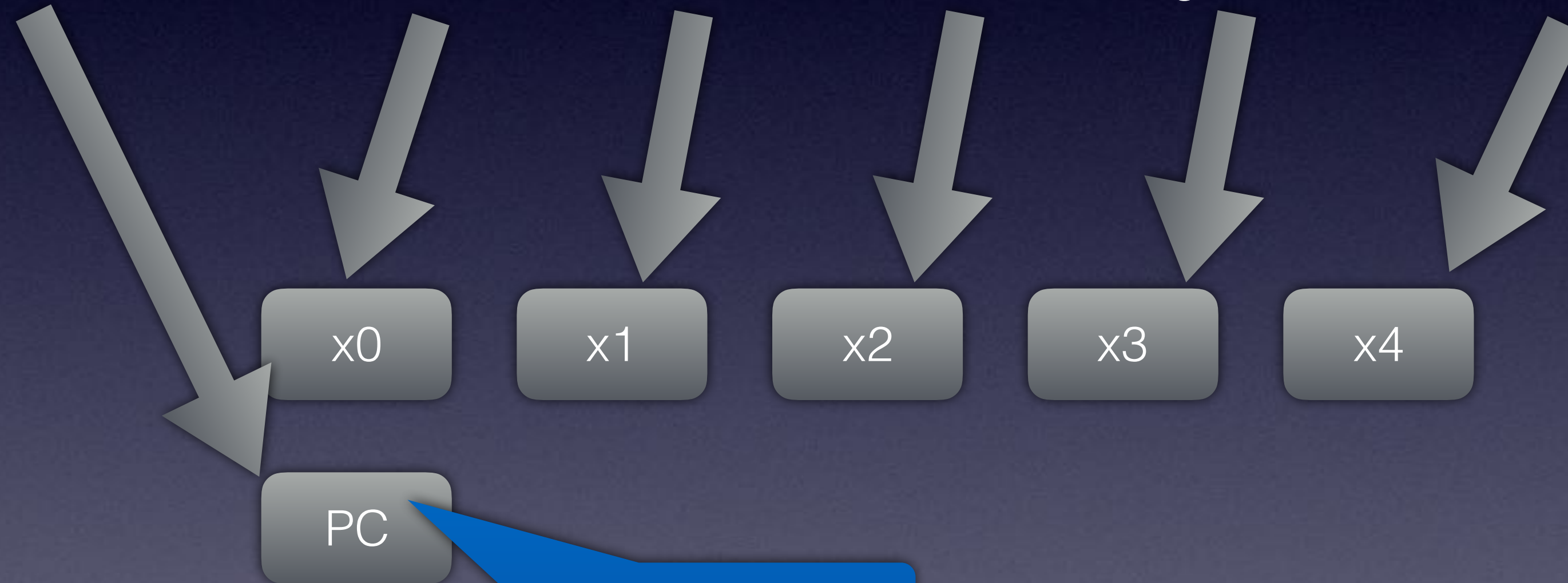
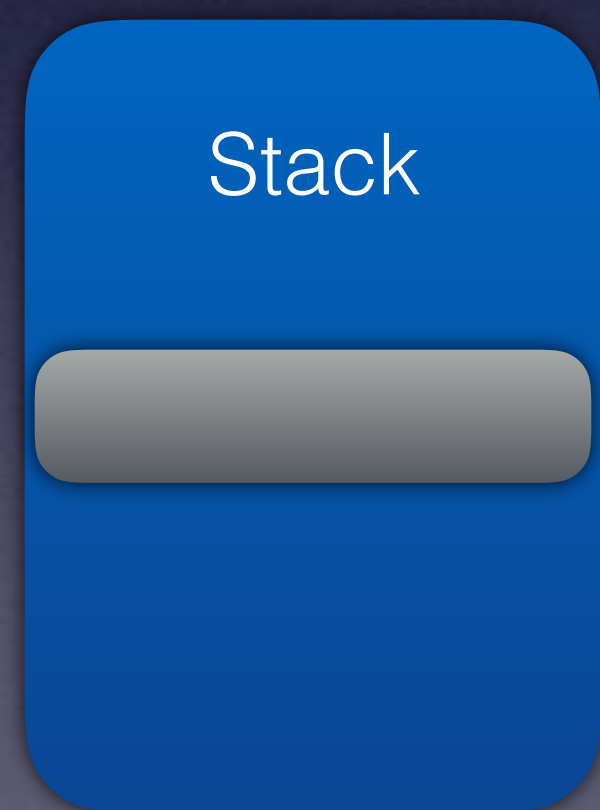


```
efi_status_t (*read)(struct block_io *this, u32 media_id, u64 lba, ulong buffer_size, void*buffer)
```





```
efi_status_t (*read)(struct block_io *this, u32 media_id, u64 lba, ulong buffer_size, void*buffer)
```





virtual

rcx

rdx

RIP

r8

r9

x0

x1

x2

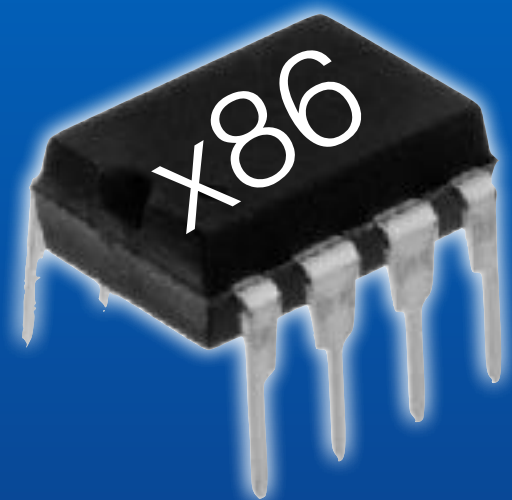
x3

x4

PC



Stack





virtual

rcx

rdx

r1p

r8

r9

x0

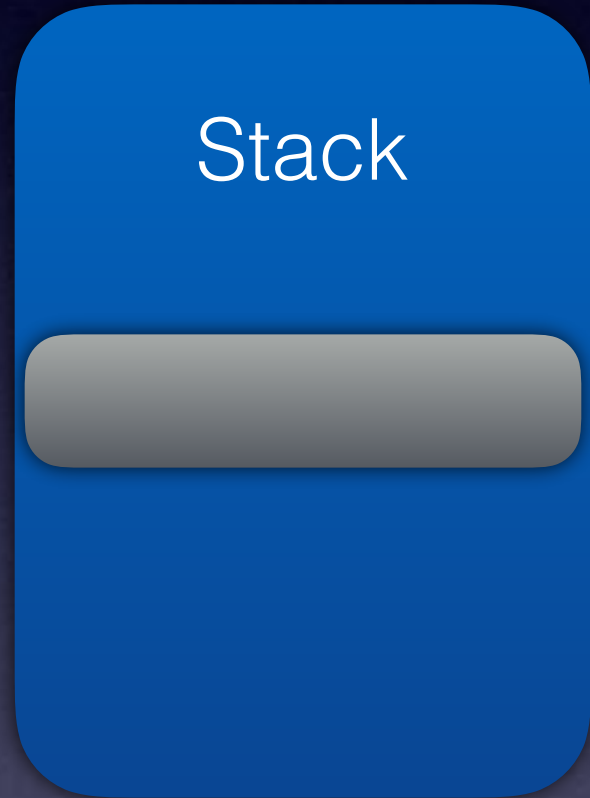
x1

x2

x3

x4

PC





virtual

rcx

rdx

r1p

r8

r9



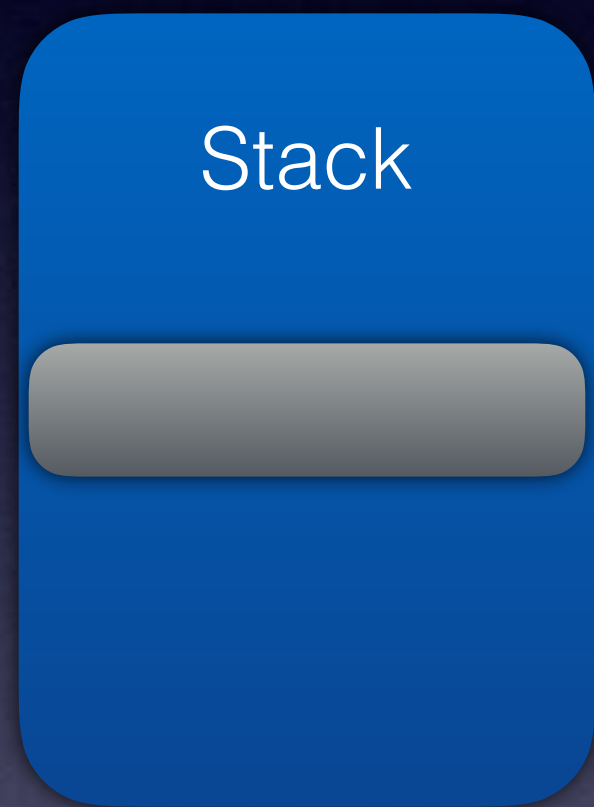
x0

x1

x2

x3

x4



PC





virtual

rcx

rdx

r1p

r8

r9

Stack

x0

x1

x2

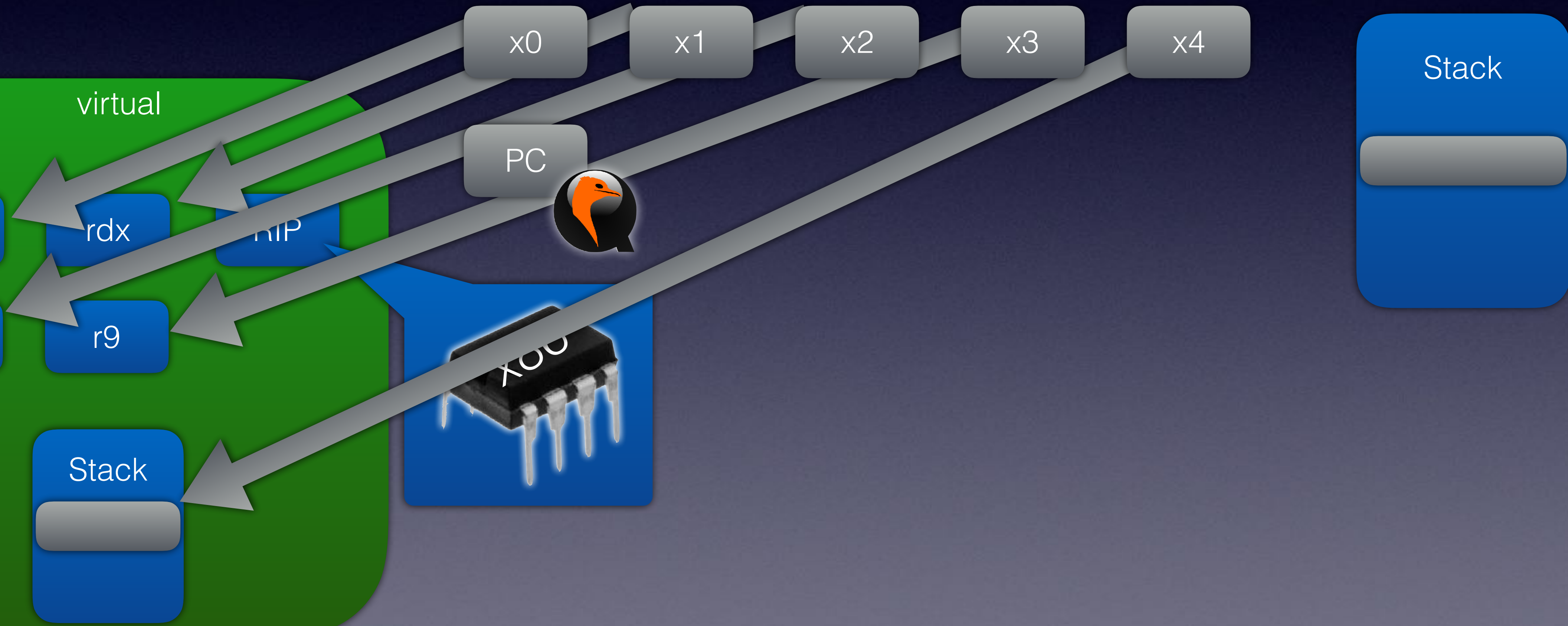
x3

x4

PC



Stack





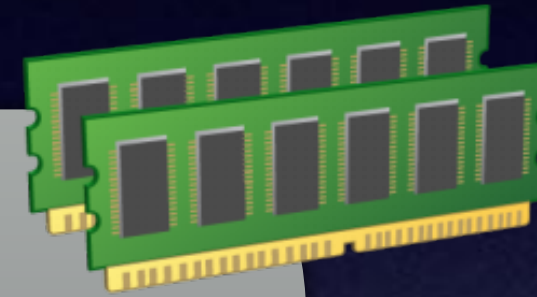
virtual





virtual

x86



ARM

x86

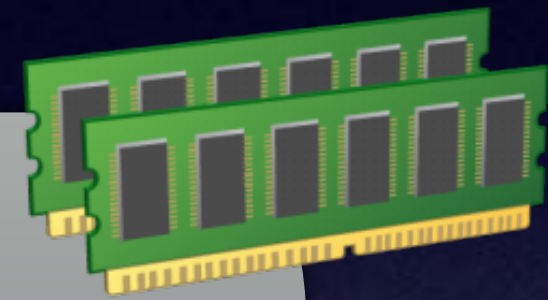
ARM

ARM



virtual

x86



ARM

x86

ARM

ARM





virtual

rcx rdx RIP

r8 r9

Stack

A green rounded rectangle representing a virtual machine. It contains several blue buttons for registers: 'rcx', 'rdx', 'RIP', 'r8', and 'r9'. At the bottom, there is a blue 'Stack' button with a grey horizontal bar below it.

x0

x1

x2

x3

x4

PC

Stack

A blue rounded rectangle representing a stack. It has the word 'Stack' at the top and a grey horizontal bar in the middle.



virtual

rcx

rdx

r8

r9

Stack

x0

x1

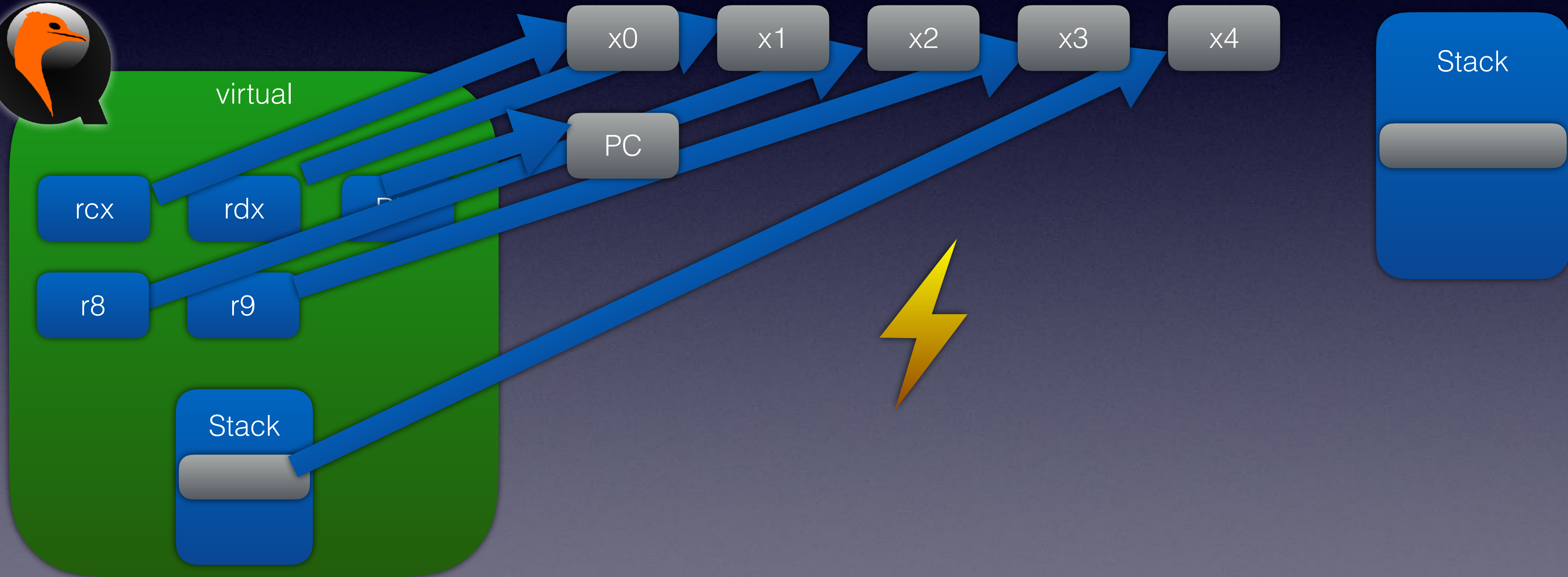
x2

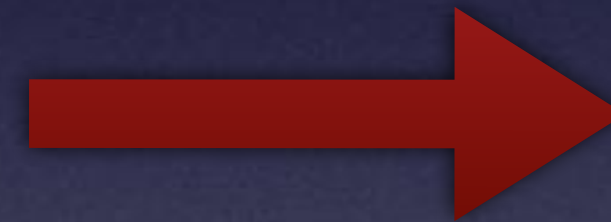
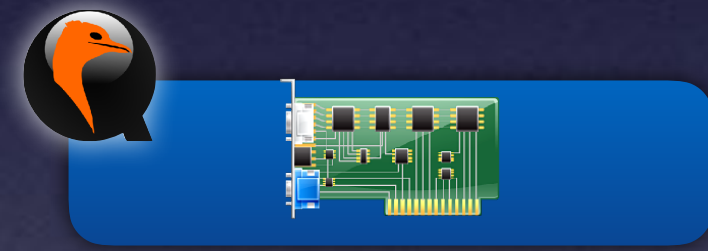
x3

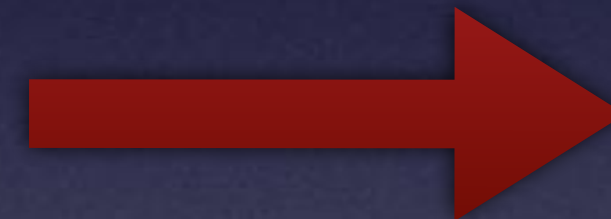
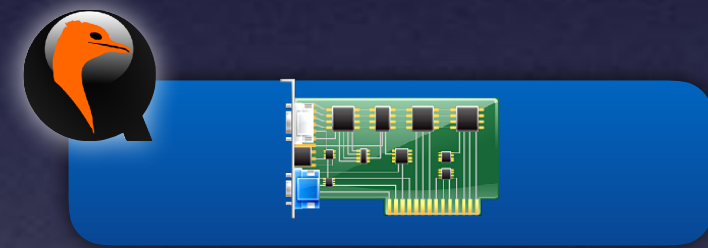
x4

PC

Stack







function level boundary



SLES 12-SP3

Advanced options for SLES 12-SP3

Start bootloader from a read-only snapshot

The highlighted entry will be executed automatically in 3s.

Why TCG

Why TCG

- C
- Isolated
- LGPL
- Supports x86_64

How to Use

How to Use

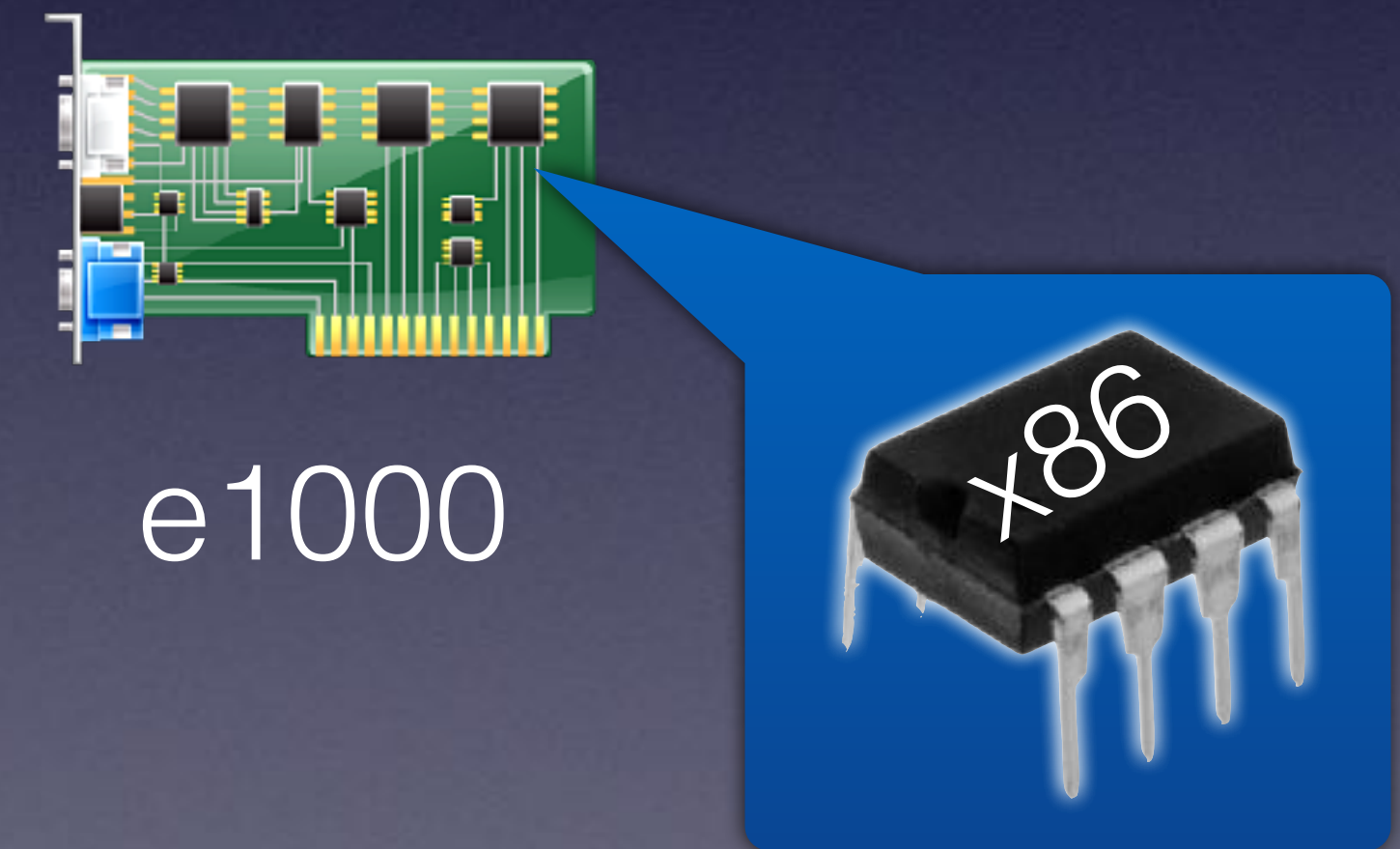
<https://github.com/ardbishesheuvel/X86EmulatorPkg.git>

How to Use

```
$ git clone https://github.com/ardbishesheuevel/edk2.git
$ cd edk2
$ git checkout origin/x86emu
$ git submodule add https://github.com/ardbishesheuevel/X86EmulatorPkg.git
$ echo "  X86EmulatorPkg/X86Emulator.inf" >> ArmVirtPkg/ArmVirtQemu.dsc
$ echo "  INF X86EmulatorPkg/X86Emulator.inf" >> ArmVirtPkg/ArmVirtQemuFvMain.fdf.inc
$ make -C BaseTools
$ . edksetup.sh
$ export GCC5_AARCH64_PREFIX=... (if you are on a non-aarch64 system)
$ build -a AARCH64 -t GCC5 -p ArmVirtPkg/ArmVirtQemu.dsc -b RELEASE
```

Demo

Demo





Backup

- EBC
- Cache Coherency

OSA Icons



Icons received from <http://www.opensecurityarchitecture.org/cms/library/icon-library>

Other Icons



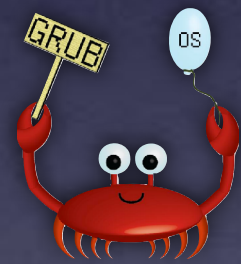
http://findicons.com/icon/202613/folder_library



<http://findicons.com/icon/download/234261/clock/128/png>



http://findicons.com/icon/439269/button_power



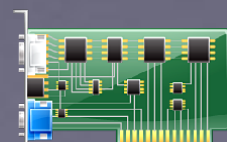
https://fosdem.org/2017/schedule/event/grub_new_maintainers/attachments/slides/1768/export/events/attachments/grub_new_maintainers/slides/1768/slides.pdf



https://de.wikipedia.org/wiki/BeagleBoard#/media/File:Beagle_Board_big.jpg



<https://thenounproject.com/term/folder-tree/27307/>



https://commons.wikimedia.org/wiki/File:Crystal_Project_Hardware.png

emojione Icons



External Sources



https://commons.wikimedia.org/wiki/File:Raspberry_Pi_B%2B_illustration.svg



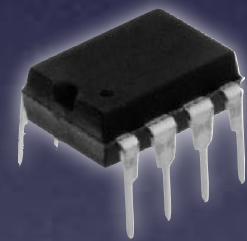
<https://commons.wikimedia.org/wiki/File:Sd-card-1377140.svg>



http://eu.mophie.com/shop/media/catalog/product/cache/3/small_image/270x330/9df78eab33525d08d6e5fb8d27136e95/u/s/usb-micro3-40-blk_usb-tip-detail_front-back_540px.jpg



<https://commons.wikimedia.org/wiki/File:Circle-icons-submarine.svg>



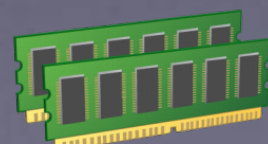
<https://commons.wikimedia.org/wiki/File:150-8-DIP.jpg>



https://commons.wikimedia.org/wiki/File:Hdd_icon.svg



https://commons.wikimedia.org/wiki/File:ARM_CPU_icon.svg



<http://findicons.com/icon/177982/memory#>