Betriebssysteme

11. Tutorium - Storage

Peter Bohner

24. Januar 2025

ITEC - Operating Systems Group

Device Management

Device Management

What kind of I/O Devices do you find in a typical system?

Device Management

What kind of I/O Devices do you find in a typical system?

- Block devices

Device Management

What kind of I/O Devices do you find in a typical system?

- Block devices
- Character devices

Device Management

What kind of I/O Devices do you find in a typical system?

- Block devices
- Character devices
- Network devices

Device Management

What kind of I/O Devices do you find in a typical system?

- Block devices
- Character devices
- Network devices

Block Devices

- Offer random access to fixed-size blocks
- Applications typically deal with a file system on top of the device
- Examples?

Device Management

What kind of I/O Devices do you find in a typical system?

- Block devices
- Character devices
- Network devices

Block Devices

- Offer random access to fixed-size blocks
- Applications typically deal with a file system on top of the device
- Examples? SSD, HDD, ...

Device Management

What kind of I/O Devices do you find in a typical system?

- Block devices
- Character devices
- Network devices

Block Devices

- Offer random access to fixed-size blocks
- Applications typically deal with a file system on top of the device
- Examples? SSD, HDD, ...

Character Devices

- Provide a stream of characters
- Examples?

Device Management

What kind of I/O Devices do you find in a typical system?

- Block devices
- Character devices
- Network devices

Block Devices

- Offer random access to fixed-size blocks
- Applications typically deal with a file system on top of the device
- Examples? SSD, HDD, ...

Character Devices

- Provide a stream of characters
- Examples? Mice, Keyboard, (classic) text terminals

Communication with Devices

Port Based I/O

Communication with Devices

Port Based I/O

Separate address space with dedicated instructions for reading/writing

- \+ Clear distinction in code $\Rightarrow$ Optimizing easier (reordering, caching, ...)
- \- Less flexible, often lower performance

Communication with Devices

Port Based I/O

Separate address space with dedicated instructions for reading/writing

- \+ Clear distinction in code ⇒ Optimizing easier (reordering, caching, ...)
- \- Less flexible, often lower performance

Memory-mapped I/O

Device registers are mapped into the physical address space. How do you access that?

Communication with Devices

Port Based I/O

Separate address space with dedicated instructions for reading/writing

+ Clear distinction in code ⇒ Optimizing easier (reordering, caching, ...)
- Less flexible, often lower performance

Memory-mapped I/O

Device registers are mapped into the physical address space. How do you access that? Normal instructions!

+ Higher flexibility: Virtual memory, larger instruction set, mostly transparent
- Some special rules apply to I/O regions software needs to be aware of

Similar Names - Compare DMA, memory-mapped I/O and memory-mapped files

DMA

Similar Names - Compare DMA, memory-mapped I/O and memory-mapped files

DMA

Direct Memory Access

Similar Names - Compare DMA, memory-mapped I/O and memory-mapped files

DMA

Direct Memory Access

- Devices can access the physical memory *without* involving the CPU

Similar Names - Compare DMA, memory-mapped I/O and memory-mapped files

DMA

Direct Memory Access

- Devices can access the physical memory *without* involving the CPU
- Needs special setup from the OS to know how and what to read/write

Memory Mapped Files

Similar Names - Compare DMA, memory-mapped I/O and memory-mapped files

DMA

Direct Memory Access

- Devices can access the physical memory *without* involving the CPU
- Needs special setup from the OS to know how and what to read/write

Memory Mapped Files

- OS abstraction: Treat a file like a normal range of virtual memory

Similar Names - Compare DMA, memory-mapped I/O and memory-mapped files

DMA

Direct Memory Access

- Devices can access the physical memory *without* involving the CPU
- Needs special setup from the OS to know how and what to read/write

Memory Mapped Files

- OS abstraction: Treat a file like a normal range of virtual memory
- No real relation to DMA, though the OS might use it to synchronize Memory Mapped Files with the underlying device

I/O Operations

How does the OS know an I/O operation is finished?

I/O Operations

How does the OS know an I/O operation is finished?

- Polling

I/O Operations

How does the OS know an I/O operation is finished?

- Polling $\Rightarrow$ Periodically check device registers

I/O Operations

How does the OS know an I/O operation is finished?

- Polling $\Rightarrow$ Periodically check device registers
- Interrupts

I/O Operations

How does the OS know an I/O operation is finished?

- Polling $\Rightarrow$ Periodically check device registers
- Interrupts $\Rightarrow$ I/O devices send an interrupt signal

Hard Disks

Hard Disk Layout

What parts can you find in a hard disk?

Hard Disk Layout

What parts can you find in a hard disk?

- Heads

Hard Disk Layout

What parts can you find in a hard disk?

- Heads
- Arms

Hard Disk Layout

What parts can you find in a hard disk?

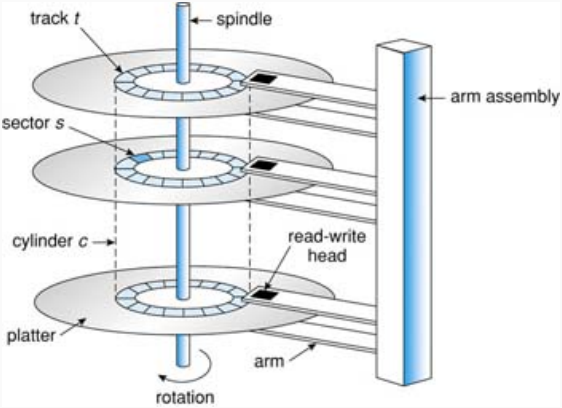
- Heads
- Arms
- Platters

Hard Disk Layout

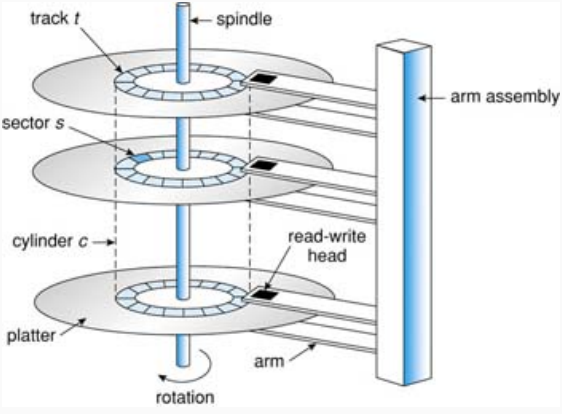
What parts can you find in a hard disk?

- Heads
- Arms
- Platters
- Spindle

Hard Disk Layout

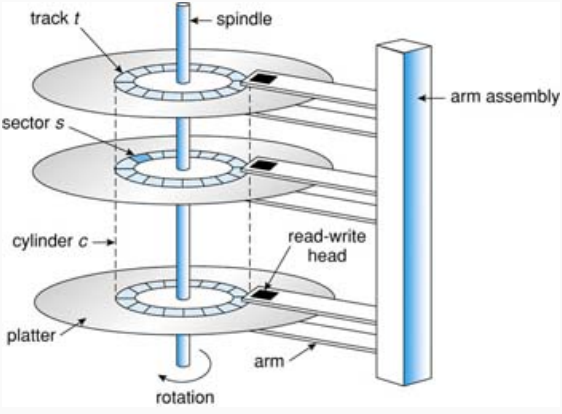


Hard Disk Layout



What do they do?

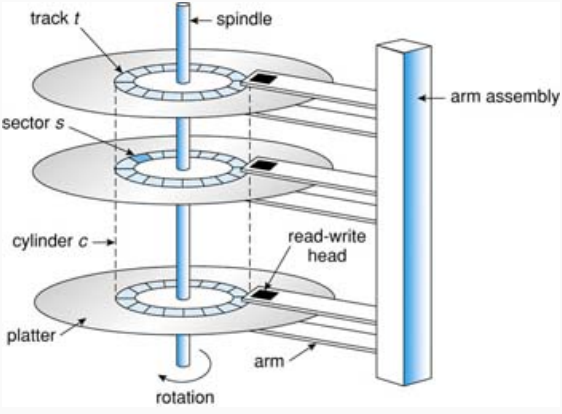
Hard Disk Layout



What do they do?

- Spindle: Spin connected platters!

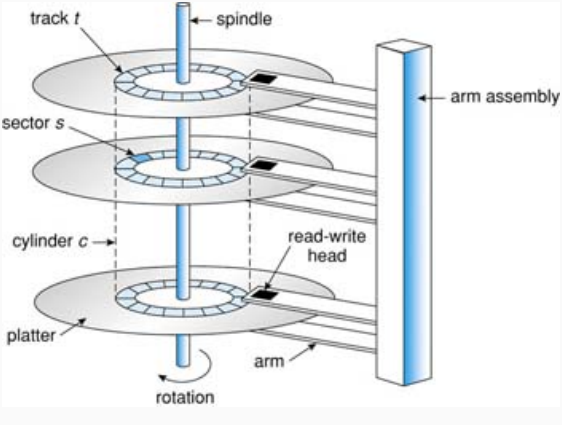
Hard Disk Layout



What do they do?

- Spindle: Spin connected platters!
- Head: Read/Write

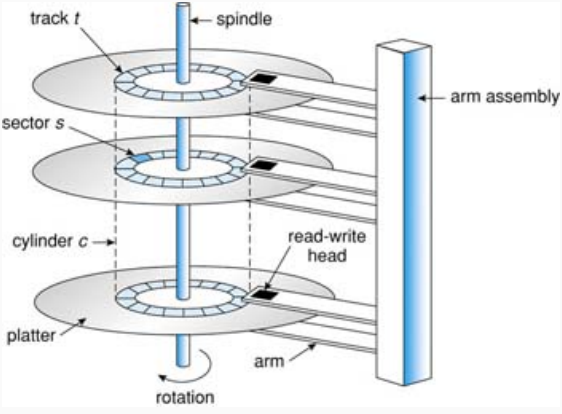
Hard Disk Layout



What do they do?

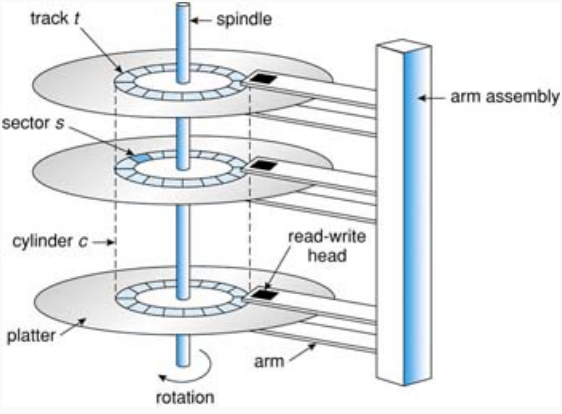
- Spindle: Spin connected platters!
- Head: Read/Write
- Arm: Move heads

Hard Disk Layout



How can you address data (512 byte blocks typically) on the disk?

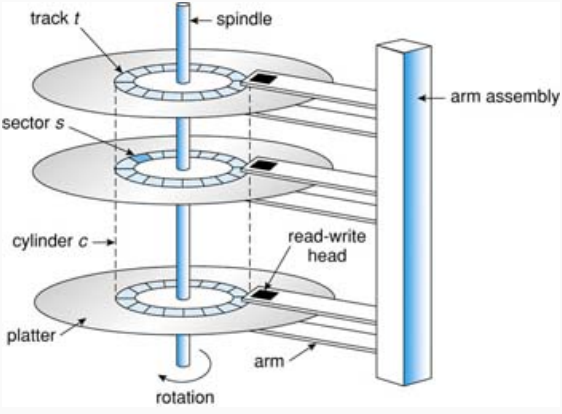
Hard Disk Layout



How can you address data (512 byte blocks typically) on the disk?

- Cylinder - Head - Sector (CHS). Limited to „small“ disks (< 8GB), rarely used these days

Hard Disk Layout



How can you address data (512 byte blocks typically) on the disk?

- Cylinder - Head - Sector (CHS). Limited to „small“ disks (< 8GB), rarely used these days
- Logical Block Addressing (LBA). Each data block has its own unique number.

How could you optimize the OS $\Leftrightarrow$ Disk interface?

Native-Command-Queuing. OS sends reads and writes in batches and (the disk | the OS) reorders them based on internal geometry.

How could you optimize the OS $\Leftrightarrow$ Disk interface?

Native-Command-Queuing. OS sends reads and writes in batches and *the disk* reorders them based on internal geometry.

Wear and Tear

What do you do when a sector is damaged?

Wear and Tear

What do you do when a sector is damaged?

Disk marks it as such and never uses it again ⇒ *Sector sparing*.

What adverse effect might this have?

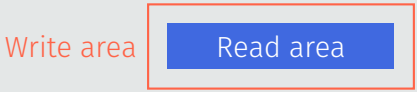
Wear and Tear

What do you do when a sector is damaged?

Disk marks it as such and never uses it again $\Rightarrow$ *Sector sparing*.
What adverse effect might this have? OS disk scheduler is unaware and optimizes for wrong geometry.

Hard Disk Layout - More Data!

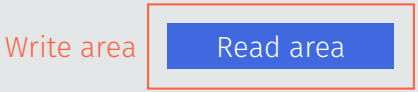
Optimizing storage



What can you do with this?

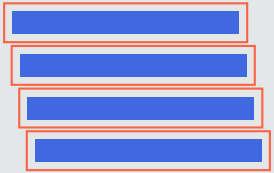
Hard Disk Layout - More Data!

Optimizing storage



What can you do with this?

Conventional layout

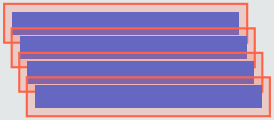


Hard Disk Layout - More Data!

Shingled Magnet Recording

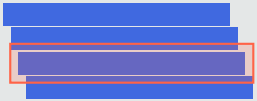
Hard Disk Layout - More Data!

Shingled Magnet Recording



Hard Disk Layout - More Data!

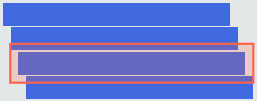
Shingled Magnet Recording



What happens when you write to this track?

Hard Disk Layout - More Data!

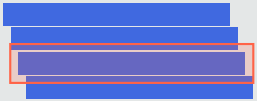
Shingled Magnet Recording



What happens when you write to this track? You overwrite the adjacent ones!

Hard Disk Layout - More Data!

Shingled Magnet Recording



What happens when you write to this track? You overwrite the adjacent ones!

⇒ Append only and group shingled tracks

Hard Disk Layout - More Data!

Shingled Magnet Recording

What happens when you write to this track? You overwrite the adjacent ones!

⇒ Append only and group shingled tracks

⇒ Can rewrite the whole group at once

Hard Disk Layout - Shingled Magnet Recording

How can such a device interface with the OS?

Hard Disk Layout - Shingled Magnet Recording

How can such a device interface with the OS?

- Pretend you are a normal disk. Buffer writes in a normal zone and flush them once they fill up a group.
  $\Rightarrow$ *Device Managed*

Hard Disk Layout - Shingled Magnet Recording

How can such a device interface with the OS?

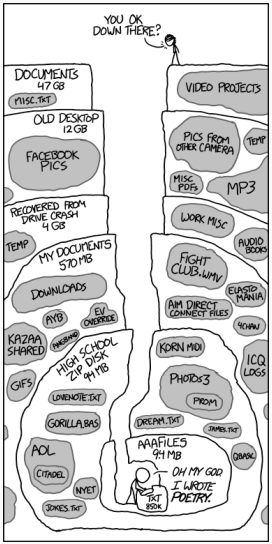
- Pretend you are a normal disk. Buffer writes in a normal zone and flush them once they fill up a group.
  $\Rightarrow$ *Device Managed*
- Tell the OS where your shingled zones are. The OS needs to write carefully to not destroy data
  $\Rightarrow$ *Host Managed*

Hard Disk Layout - Shingled Magnet Recording

How can such a device interface with the OS?

- Pretend you are a normal disk. Buffer writes in a normal zone and flush them once they fill up a group.
  $\Rightarrow$ *Device Managed*
- Tell the OS where your shingled zones are. The OS needs to write carefully to not destroy data
  $\Rightarrow$ *Host Managed*
- Compromise. Tell the OS where your singled zones are and expose their tail. If the OS writes to the tail, directly commit it - else buffer.
  $\Rightarrow$ *Host Aware*

Ende



XKCD 1360 - Old Files

F R A G E N?

https://forms.gle/9CwJSKidKibubran9

Bis nächste Woche