

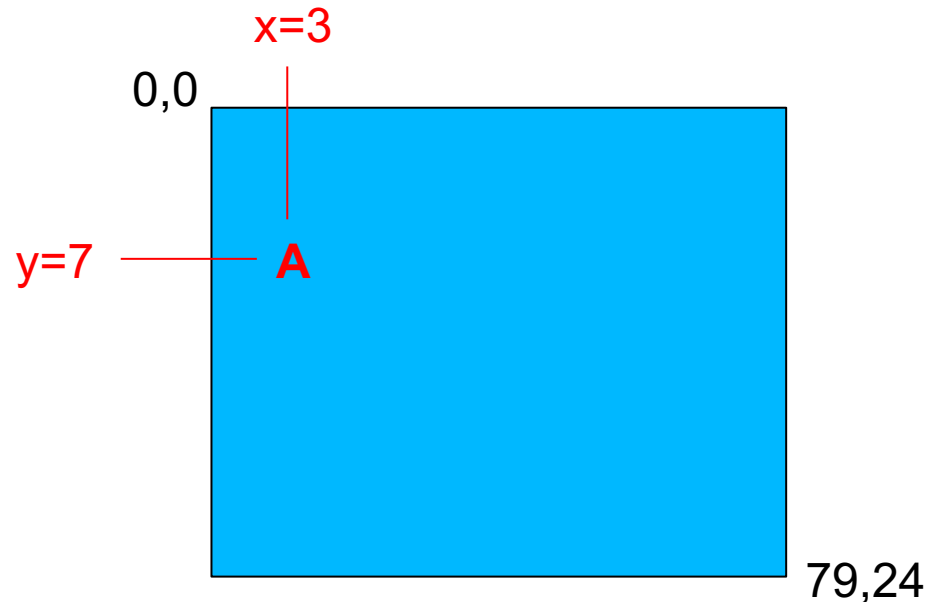
CGA-Programmierung

Ausgabestrom

- `print!` und `println!`
 - Standard-Makros für Ausgaben in Rust
 - Realisiert mithilfe eines `Writers` in `cga_print.rs`
 - Die Umwandlung von Zahlen nach Hexadezimal- oder Binäreformat erfolgt automatisch in `core::fmt`
 - Die Ausgabe des Zeichenstroms erfolgt mithilfe der Funktion `cga::print_byte(byte)` (an der aktuellen Text-Cursor-Position)

Text-Cursor

- Die Position ist ein 16 Bit Offset zur linken oberen Ecke
- Die Textauflösung ist 80x25 Zeichen
- Cursor-Position im Beispiel = (3,7)
- Offset = $3 * 80 + 7 = 247$



Text-Cursor

- Der 16 Bit Offset wird in folgende Index-Register geschrieben / gelesen
- Jedes Indexregister kann nur ein Byte schreiben / lesen
- Die Auswahl eines Indexregisters 14 oder 15 erfolgt über Port 0x3d4
- Die Bytes werden über das Datenregister geschrieben / gelesen

Port	Register	Zugriffsart
3d4	Indexregister	nur schreiben
3d5	Datenregister	lesen und schreiben

Index	Register	Bedeutung
14	Cursor (high)	Zeichenoffset der Cursorposition
15	Cursor (low)	

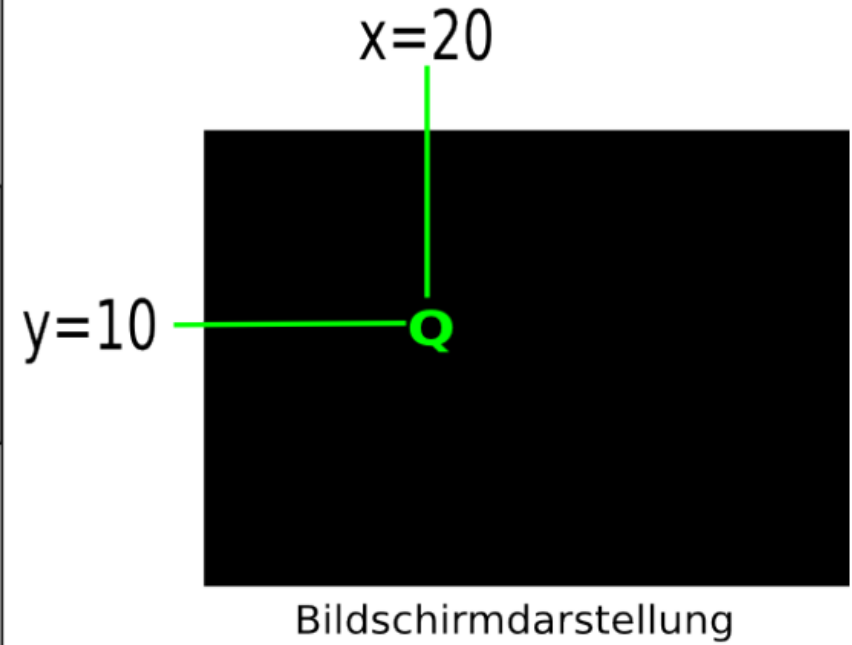
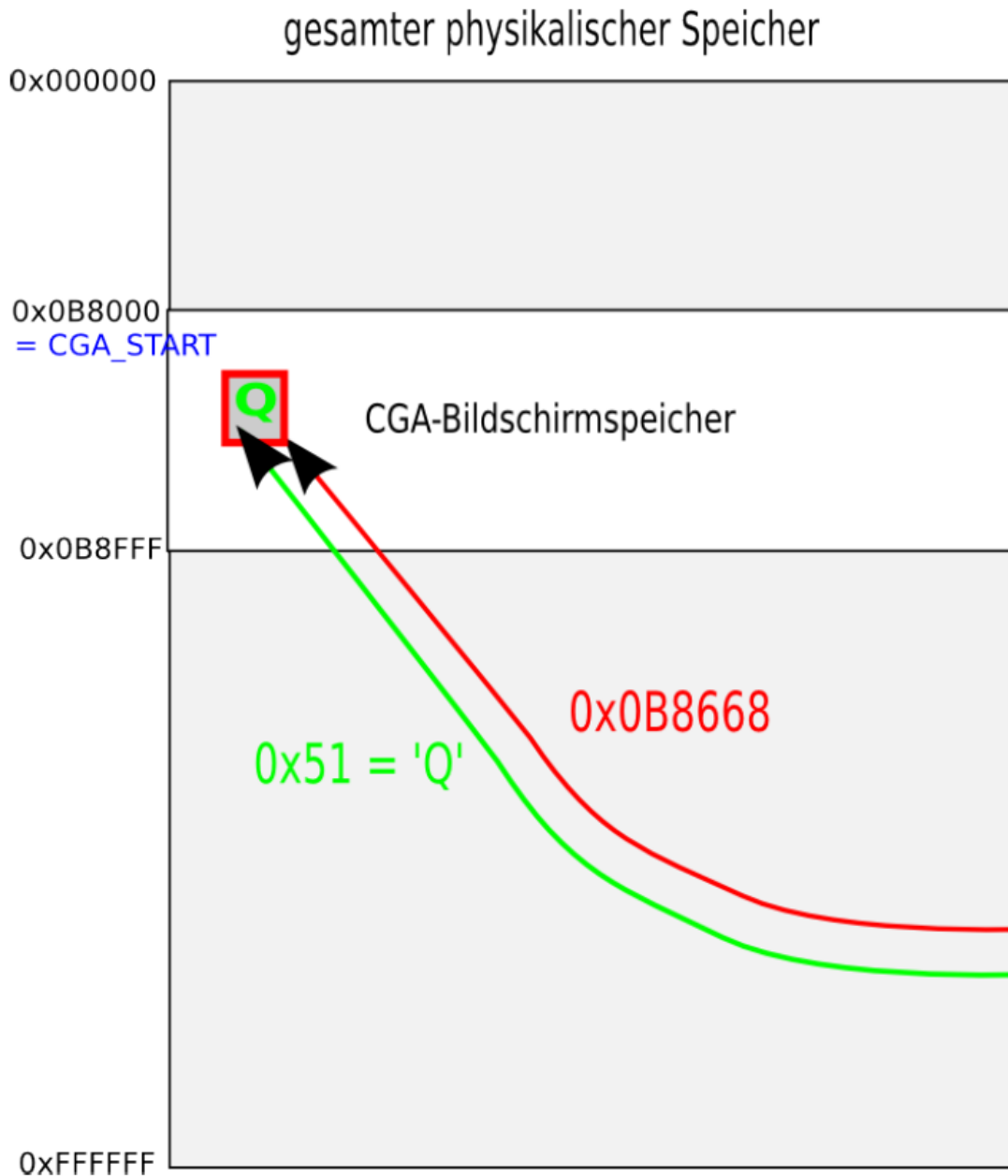
Anzeige von Zeichen

- Wird von `print_byte` gerufen
- `show(x, y, c, attrib)`
 - Zeichen `c` mit Attribut `attrib` an Position `x`, `y`

```
pub const CGA_START: u64 = 0xb8000;  
  
// ...  
  
let pos = CGA_START + 2*(x + y*80);  
  
unsafe {  
    *(pos as *mut u8) = 'Q' as u8;  
}
```

- Fehlt hier noch etwas?

Anzeige von Zeichen



```
let x = 20;  
let y = 20;  
let pos = CGA_START + 2*(x + y*80);  
  
unsafe {  
    *(pos as *mut u8) = 'Q' as u8;  
}
```

Anzeige von Zeichen

- Je zwei Bytes im Bildspeicher pro Bildposition!
- Gerade Adressen: ASCII-Code
- Ungerade Adressen: Attributbyte

```
pub const CGA_START: u64 = 0xb8000;  
  
// ...  
  
let pos = CGA_START + 2*(x + y*80);  
  
unsafe {  
    *(pos as *mut u8) = 'Q' as u8;  
    *(pos+1 as *mut u8) = 0xF; // weiss auf schwarz  
}
```

Attribut-Byte

- Zu jedem Zeichen können die Merkmale Vordergrundfarbe, Hintergrundfarbe und Blinken einzeln festgelegt werden.
- Für diese Attribute steht pro Zeichen ein Byte zur Verfügung,

Darstellungsattribute	
Bits 0-3	Vordergrundfarbe
Bits 4-6	Hintergrundfarbe
Bit 7	Blinken

Attribut-Byte

- Im CGA-Textmodus stehen die folgenden 16 Farben zur Verfügung:

Farbpalette			
0	Schwarz	8	Dunkelgrau
1	Blau	9	Hellblau
2	Grün	10	Hellgrün
3	Cyan	11	Hellcyan
4	Rot	12	Hellrot
5	Magenta	13	Hellmagenta
6	Braun	14	Gelb
7	Hellgrau	15	Weiß

- Da für die Hintergrundfarbe im Attributbyte nur drei Bits zur Verfügung stehen, können auch nur die ersten acht Farben zur Hintergrundfarbe gewählt werden.

Weiterführende Informationen

- Wer mehr zum Thema VGA-Grafikkarten-Programmierung lesen möchte, sei auf das FreeVGA-Projekt verwiesen:

<http://www.osdever.net/FreeVGA/home.htm>

- Ist nicht notwendig für unsere Aufgabe!