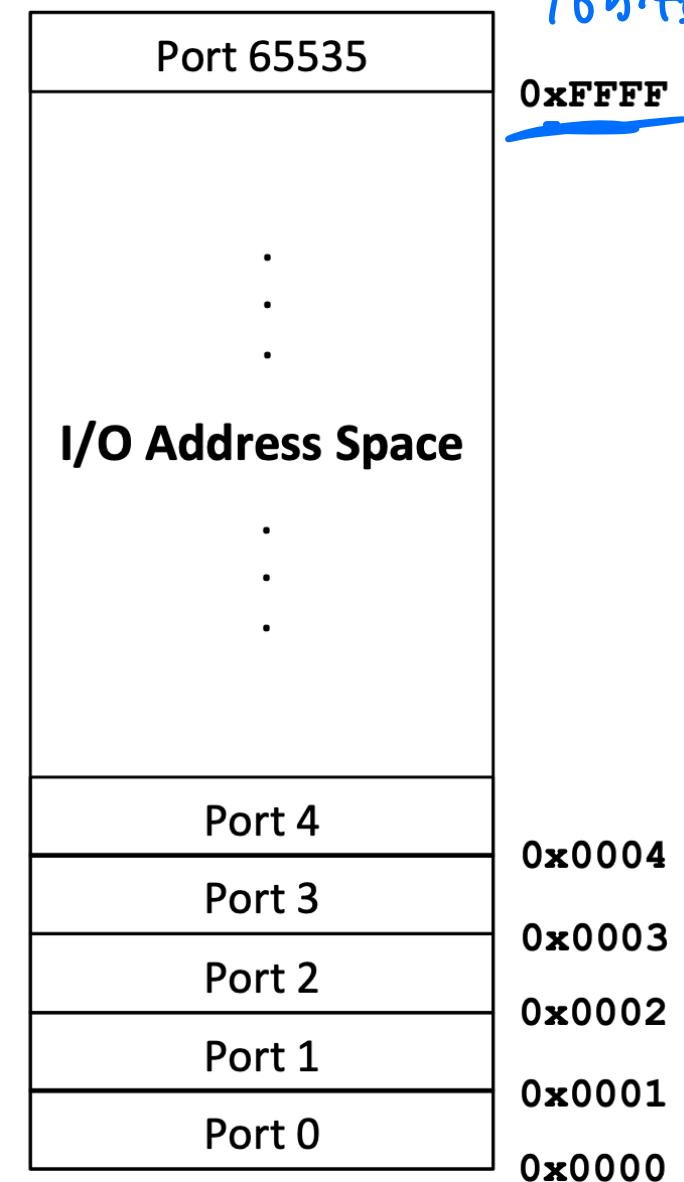


# Port I/O Address Space

- Software and hardware architectures of x86 architecture support a separate address space called “I/O Address Space”
  - Separate from memory space
- Access to this separate I/O space is handled through a set of I/O instructions
  - IN, OUT, INS, OUTS
- Access requires Ring0 privileges
  - Access requirement does not apply to all operating modes (like Real-Mode)
- The processor allows 64 KB+3 bytes to be addressed within the I/O space
- Harkens back to a time when memory was not so plentiful
- You may never see port I/O when analyzing high-level applications, but in systems programming (and especially BIOS) you will see lots of port I/O
- One of the biggest impediments to understanding what's going on in a BIOS



1. Device drivers
2. Mechanics of communication
3. Demo: implementing a tty dev (egos-NU)
4. Communication configurations

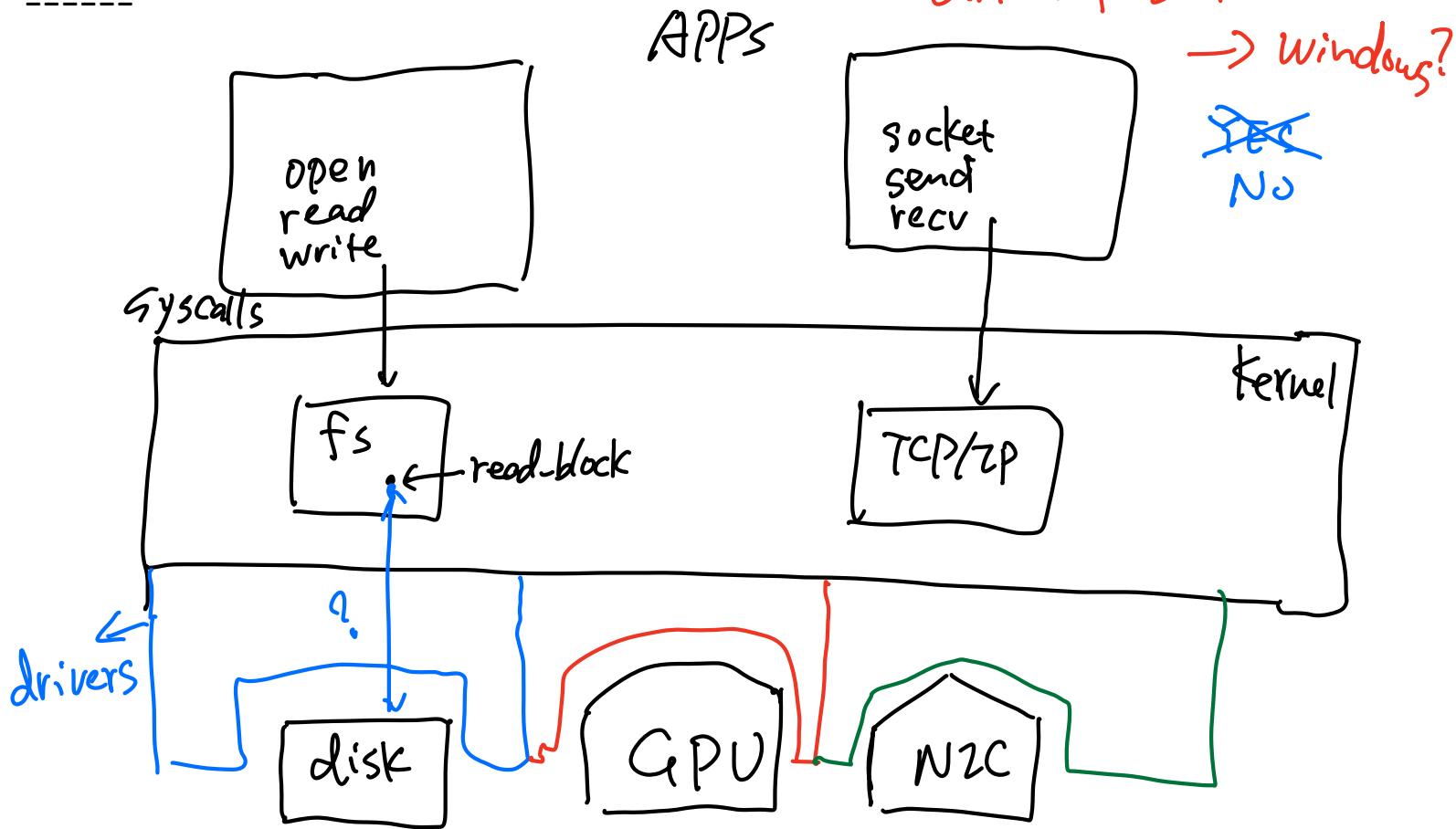
Q: ① NVIDIA GPU.

driver → AMD GPU?

② NVidia GPU  
driver for Linux

→ Windows?

~~Yes~~  
No



## 2. Mechanics of communication between CPU and I/O devices

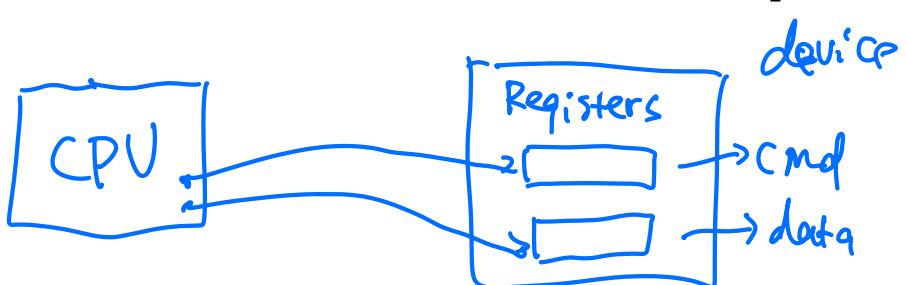
- [ ] (a) explicit I/O instructions ✘
- [ ] (b) memory-mapped I/O ↙
- [ ] (c) interrupts
- [ ] (d) through memory: both CPU and the device see the same memory

DMA

x86:

op: inb, outb, inw, outw  
    ↳ 1B           ↳ word  
    ↳ 2B

operands: ZO address



```

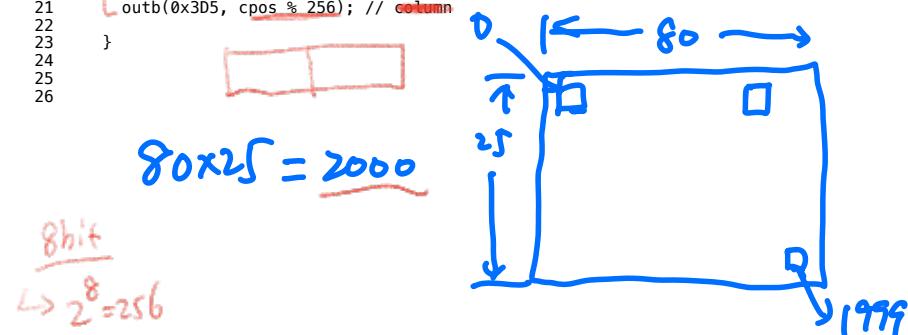
1 CS3650: select, synchronous I/O multiplexing
2
3 1. select interfaces
4
5 a) select
6
7 * int select(int nfds, fd_set *restrict readfds,
8     fd_set *restrict writefds, fd_set *restrict errorfds,
9     struct timeval *restrict timeout);
10
11 select() examines the I/O descriptor sets whose addresses are passed in
12 readfds, writefds, and errorfds to see if some of their descriptors are
13 ready for reading, are ready for writing, or have an exceptional
14 condition pending, respectively.
15
16 b) fd_set manipulation
17
18 * FD_ZERO(fd_set *set);           Clear all entries from the set.
19 * FD_SET(int fd, fd_set *set);    Add fd to the set.
20 * FD_CLR(int fd, fd_set *set);   Remove fd from the set.
21 * FD_ISSET(int fd, fd_set *set); Return true if fd is in the set.
22
23
24 2. An example - a chat server
25
26 // Below is a code snippet using select()
27
28 int fds[2] = {0, 0};
29 fds[0] = ...           // socket connection 1
30 fds[1] = ...           // socket connection 2
31
32 fd_set readfds;
33
34 while(1) {
35     FD_ZERO(&readfds);
36     for (int i=0; i<2; i++) {
37         FD_SET(fds[i], &readfds);
38     }
39
40     int maxfd = ...          // Q: what is the maxfd?
41
42     select(maxfd+1, &readfds, NULL, NULL, NULL);
43
44     for (int i=0; i<2; i++) {
45         if (FD_ISSET(fds[i], &readfds)) {
46             print(fds[i], ...); // print msg received
47         }
48     }
49
50 } ... // wrap up and exit

```

```

1 CS3650: I/O and device driver
2
3 1. An example of I/O instructions:
4     Setting the cursor position
5
6 The code below is excerpted from WeensyOS's k-hardware.c. It
7 uses I/O instructions to set a blinking cursor in the upper left of
8 the screen.
9
10 // console_show_cursor(cpos)
11 // Move the console cursor to position 'cpos',
12 // which should be between 0 and 80 * 25.
13
14 void console_show_cursor(int cpos) {
15     if (cpos < 0 || cpos > CONSOLE_ROWS * CONSOLE_COLUMNS)
16         cpos = 0;
17
18     outb(0x3D4, 14); // Command 14 = upper byte of position
19     outb(0x3D5, cpos / 256); // row
20     outb(0x3D4, 15); // Command 15 = lower byte of position
21     outb(0x3D5, cpos % 256); // column
22
23 }
24
25
26

```



```

27
28 2. Memory-mapped I/O
29
30 a. Here is a 32-bit PC's physical memory map:
31
32 +-----+ <- 0xFFFFFFFF (4GB)
33 | 32-bit
34 | memory mapped
35 | devices
36 |
37 | \/\ / \ / \ / \ / \ / \
38 |
39 | Unused
40 |
41 | +-----+ <- depends on amount of RAM
42 |
43 | Extended Memory
44 |
45 | BIOS ROM <- 0x00100000 (1MB)
46 | 16-bit devices,
47 | expansion ROMs
48 | VGA Display <- 0x000C0000 (768KB)
49 | Low Memory <- 0x000A0000 (640KB)
50 |
51 | +-----+ <- 0x000F0000 (960KB)
52 |
53 | [ ] 0xB8000
54 |
55 | [ ] 0xB8000
56 |
57 | [ ] 0xB8000
58 | [ ] 0xB8000
59 | [ ] 0xB8000
60 |
61 [Credit to Frans Kaashoek, Robert Morris, and
62 Nickolai Zeldovich for this picture]
63
64

```

Diagram of a 32-bit PC's physical memory map:

- Top level: 32-bit memory mapped devices (0xFFFFFFFF)
- Second level: Unused, depends on amount of RAM, Extended Memory.
- Third level: BIOS ROM (1MB), 16-bit devices, expansion ROMs, VGA Display (768KB), Low Memory (640KB).
- Fourth level: BIOS ROM (0x00100000), 16-bit devices, expansion ROMs (0x000F0000), VGA Display (0x000C0000), Low Memory (0x000A0000).
- Bottom level: Low Memory (0x00000000) to 0xB8000.

Annotations:

- Red bracket on the left side of the diagram.
- Red arrow pointing down from the bottom of the diagram to the address 0xB8000.
- Handwritten notes: "move A,B", "A address → read", "B add res → write".
- Address 0xB8000 highlighted in red.

```

65
66 b. Loads and stores to the device memory "go to hardware".
67
68 An example is in the console printing code from WeensyOS.
69 Here is an excerpt from link/shared.ld:
70
71 /* Compare the address below to the map above. */
72 PROVIDE(console = 0x88000);
73
74 This is an excerpt from lib.c; notice how it uses the address
75 "console":
76
77 /*
78 * prints a character to the console at the specified
79 * cursor position in the specified color.
80 * Question: what is going on in the check
81 * if (c == '\n')
82 * ?
83 * Hint: '\n' is "C" for "newline" (the user pressed enter).
84 */
85 static void console_putc(console_printer* p, unsigned char c, int color) {
86     console_printer* cp = (console_printer*) p;
87     if (cp->cursor >= console + CONSOLE_ROWS * CONSOLE_COLUMNS) {
88         cp->cursor = console;
89     }
90     if (c == '\n') {
91         int pos = (cp->cursor - console) % 80;
92         for (; pos != 80; pos++) {
93             *cp->cursor++ = ' ' | color;
94         }
95     } else {
96         *cp->cursor++ = c | color;
97     }
98 }

```

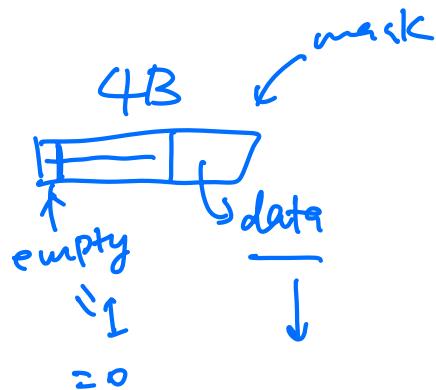
Diagram illustrating memory access:

- Diagram shows a memory buffer with rows labeled 0xB8000, 0xB8000, 0xB8000, 0xB8000, and 0xB8000.
- Diagram shows a cursor position at row 25, column 80.
- Diagram shows a character 'e' being written at position (25, 80).
- Diagram shows a character ' ' (space) being written at positions (25, 1), (25, 2), ..., (25, 79).
- Diagram shows a character 'e' being written at position (26, 1).
- Diagram shows a character ' ' (space) being written at positions (26, 2), ..., (26, 79).
- Diagram shows a character 'e' being written at position (27, 1).
- Diagram shows a character ' ' (space) being written at positions (27, 2), ..., (27, 79).
- Diagram shows a character 'e' being written at position (28, 1).
- Diagram shows a character ' ' (space) being written at positions (28, 2), ..., (28, 79).
- Diagram shows a character 'e' being written at position (29, 1).
- Diagram shows a character ' ' (space) being written at positions (29, 2), ..., (29, 79).
- Diagram shows a character 'e' being written at position (30, 1).
- Diagram shows a character ' ' (space) being written at positions (30, 2), ..., (30, 79).
- Diagram shows a character 'e' being written at position (31, 1).
- Diagram shows a character ' ' (space) being written at positions (31, 2), ..., (31, 79).
- Diagram shows a character 'e' being written at position (32, 1).
- Diagram shows a character ' ' (space) being written at positions (32, 2), ..., (32, 79).
- Diagram shows a character 'e' being written at position (33, 1).
- Diagram shows a character ' ' (space) being written at positions (33, 2), ..., (33, 79).
- Diagram shows a character 'e' being written at position (34, 1).
- Diagram shows a character ' ' (space) being written at positions (34, 2), ..., (34, 79).
- Diagram shows a character 'e' being written at position (35, 1).
- Diagram shows a character ' ' (space) being written at positions (35, 2), ..., (35, 79).
- Diagram shows a character 'e' being written at position (36, 1).
- Diagram shows a character ' ' (space) being written at positions (36, 2), ..., (36, 79).
- Diagram shows a character 'e' being written at position (37, 1).
- Diagram shows a character ' ' (space) being written at positions (37, 2), ..., (37, 79).
- Diagram shows a character 'e' being written at position (38, 1).
- Diagram shows a character ' ' (space) being written at positions (38, 2), ..., (38, 79).
- Diagram shows a character 'e' being written at position (39, 1).
- Diagram shows a character ' ' (space) being written at positions (39, 2), ..., (39, 79).
- Diagram shows a character 'e' being written at position (40, 1).
- Diagram shows a character ' ' (space) being written at positions (40, 2), ..., (40, 79).
- Diagram shows a character 'e' being written at position (41, 1).
- Diagram shows a character ' ' (space) being written at positions (41, 2), ..., (41, 79).
- Diagram shows a character 'e' being written at position (42, 1).
- Diagram shows a character ' ' (space) being written at positions (42, 2), ..., (42, 79).
- Diagram shows a character 'e' being written at position (43, 1).
- Diagram shows a character ' ' (space) being written at positions (43, 2), ..., (43, 79).
- Diagram shows a character 'e' being written at position (44, 1).
- Diagram shows a character ' ' (space) being written at positions (44, 2), ..., (44, 79).
- Diagram shows a character 'e' being written at position (45, 1).
- Diagram shows a character ' ' (space) being written at positions (45, 2), ..., (45, 79).
- Diagram shows a character 'e' being written at position (46, 1).
- Diagram shows a character ' ' (space) being written at positions (46, 2), ..., (46, 79).
- Diagram shows a character 'e' being written at position (47, 1).
- Diagram shows a character ' ' (space) being written at positions (47, 2), ..., (47, 79).
- Diagram shows a character 'e' being written at position (48, 1).
- Diagram shows a character ' ' (space) being written at positions (48, 2), ..., (48, 79).
- Diagram shows a character 'e' being written at position (49, 1).
- Diagram shows a character ' ' (space) being written at positions (49, 2), ..., (49, 79).
- Diagram shows a character 'e' being written at position (50, 1).
- Diagram shows a character ' ' (space) being written at positions (50, 2), ..., (50, 79).
- Diagram shows a character 'e' being written at position (51, 1).
- Diagram shows a character ' ' (space) being written at positions (51, 2), ..., (51, 79).
- Diagram shows a character 'e' being written at position (52, 1).
- Diagram shows a character ' ' (space) being written at positions (52, 2), ..., (52, 79).
- Diagram shows a character 'e' being written at position (53, 1).
- Diagram shows a character ' ' (space) being written at positions (53, 2), ..., (53, 79).
- Diagram shows a character 'e' being written at position (54, 1).
- Diagram shows a character ' ' (space) being written at positions (54, 2), ..., (54, 79).
- Diagram shows a character 'e' being written at position (55, 1).
- Diagram shows a character ' ' (space) being written at positions (55, 2), ..., (55, 79).
- Diagram shows a character 'e' being written at position (56, 1).
- Diagram shows a character ' ' (space) being written at positions (56, 2), ..., (56, 79).
- Diagram shows a character 'e' being written at position (57, 1).
- Diagram shows a character ' ' (space) being written at positions (57, 2), ..., (57, 79).
- Diagram shows a character 'e' being written at position (58, 1).
- Diagram shows a character ' ' (space) being written at positions (58, 2), ..., (58, 79).
- Diagram shows a character 'e' being written at position (59, 1).
- Diagram shows a character ' ' (space) being written at positions (59, 2), ..., (59, 79).
- Diagram shows a character 'e' being written at position (60, 1).
- Diagram shows a character ' ' (space) being written at positions (60, 2), ..., (60, 79).
- Diagram shows a character 'e' being written at position (61, 1).
- Diagram shows a character ' ' (space) being written at positions (61, 2), ..., (61, 79).
- Diagram shows a character 'e' being written at position (62, 1).
- Diagram shows a character ' ' (space) being written at positions (62, 2), ..., (62, 79).
- Diagram shows a character 'e' being written at position (63, 1).
- Diagram shows a character ' ' (space) being written at positions (63, 2), ..., (63, 79).
- Diagram shows a character 'e' being written at position (64, 1).
- Diagram shows a character ' ' (space) being written at positions (64, 2), ..., (64, 79).

Demo: How to implement a tty device (a terminal device using UART protocol)?

Base Address : 0x100(3000)

offset : 0x04 rxdata

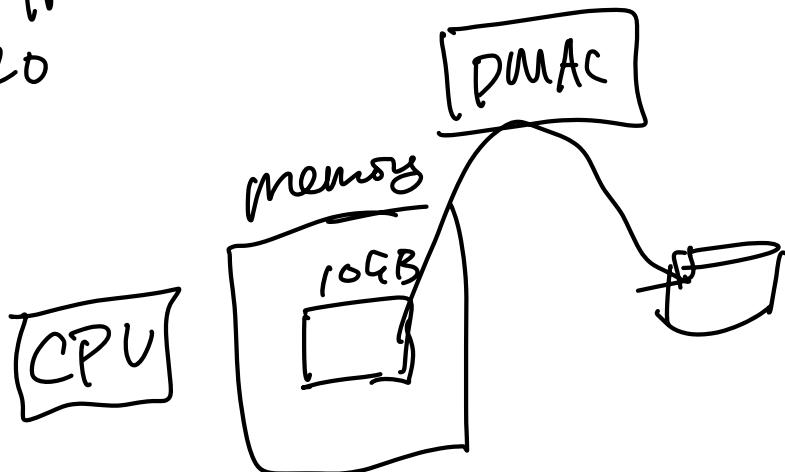


- Configurations.

- Polling & interrupt

interrupt X DMA

- P20 & DMA
  - ↳ Port-mapped 70 ← 70 address
  - ↳ MM20



{Polling, interrupt} X {P20, DMA}