Hobbyist APLing in the 21st Century.

Alex Weiner (New Jersey, USA)
Alex Weiner
Alex Weiner

- Electrical & Computer Engineer
Alex Weiner

- Electrical & Computer Engineer
- APLer
Why APL?

- Lots of things can be modeled as a matrix
Why APL?

- Lots of things can be modeled as a matrix
  - Circuits
  - Images
  - Differential equations
Why APL?

- Lots of things can be modeled as a matrix
Why APL?

- Lots of things can be modeled as a matrix
  - Circuits
  - Images
  - Differential equations
Why APL?

- Lots of things can be modeled as a matrix
  - Circuits
  - Images
  - Differential equations
- Computer Architecture
Why APL?

● Lots of things can be modeled as a matrix
  ○ Circuits
  ○ Images
  ○ Differential equations

● Computer Architecture
  ○ “A Programming Language” By Iverson
“Hobbyist APLing in the 21st Century”

What does that even mean?
Hobbyist

- Must be fun
Hobbyist

- Must be fun
- Must be low cost
Hobbyist

- Must be fun
- Must be low cost
- Can be non-practical
Hobbyist

- Must be fun
- Must be low cost
- Can be non-practical
- Must not be frustrating
Hobbyist

- Must be fun
- Must be low cost
- Can be non-practical
- Must not be frustrating
- A learning experience
APLing

- Programming in any APL dialect or related language
APLing

- Programming in any APL dialect or related language
- Talking about any APL dialect or related language
21st Century

- A web application
21st Century

● A web application
  ○ Your program ultimately outputs HTML
Application development in APL
Pick an operating system
Pick an operating system

- Linux
- macOS
- Windows
Pick an APL
Pick an APL

- APL
  - Dyalog APL
  - GNU APL
  - J
Pick an APL

- APL
  - Dyalog APL
  - GNU APL
  - J

- Other
  - J
  - A+,Q,K
  - APL2000, NARS2000
  - ELI
  - S (R is S!)
  - MATLAB, Mathematica
  - Go, C++ (iota)
Determine an application
Determine an application

- A hard problem in computer programming
Determine an application

- A hard problem in computer programming
- Do something cool
Determine an application

- A hard problem in computer programming
- Do something cool
- Something that shows my friends how cool APL is
My Application
My Application

- Called “Flake”
My Application

- Called “Flake”
  - Chops
  - Matches
My Application

- Called “Flake”
  - Chops
  - Matches
- Examples
Currently online: YHNJMUIK.COM
Start Developing!
Requirements
Requirements

● Image Code
Requirements

- Image Code
  - Into APL
  - Out of APL
Requirements

- Image Code
  - Into APL
  - Out of APL
- Web Code
Requirements

- Image Code
  - Into APL
  - Out of APL

- Web Code
  - Output valid HTML
  - Parse form data as input
Requirements

● Image Code
  ○ Into APL
  ○ Out of APL

● Web Code
  ○ Output valid HTML
  ○ Parse form data as input

● Flake Code
Requirements

IMG ➔ APL ➔ WEB

IMG ← APL ← WEB
Those arrows are written in APL, also ↑
Architecture (Version 1)
Architecture (Version 1)

- PHP
Architecture (Version 1)

- PHP
  - Bash
Architecture (Version 1)

- PHP
  - Bash
    - J
Architecture (Version 1)

- PHP
  - Bash
  - J

A TOTAL MESS!
Write every part in APL
Start Developing!
Start Developing! (almost)
Set Up the Keyboard
Set Up the Keyboard

- This is not an APL problem
Set Up the Keyboard

- This is not an APL problem
  - This is an OS problem
Set Up the Keyboard

- This is not an APL problem
  - This is an OS problem
  - Experienced by a large number of APLers
Set Up the Keyboard

- This is not an APL problem
  - This is an OS problem
  - Experienced by a large number of APLers
- Unicode
Set Up the Keyboard

- This is not an APL problem
  - This is an OS problem
  - Experienced by a large number of APLers
- Unicode
  - The APL portion in “Miscellaneous Technical”
Set Up the Keyboard

sudo apt-get install subversion

sudo **svn co** [http://svn.savannah.gnu.org/svn/apl/trunk](http://svn.savannah.gnu.org/svn/apl/trunk)

cd trunk

xmodmap support-files/Dyalog-Keyboard/apl.xmodmap-alexweiner
Set Up the Keyboard

- Files about the keyboard
  - trunk/README-3-keyboard
  - trunk/support-files/Dyalog-Keyboard/README
Start Developing
Start Developing! (really!)
How to write a Bitmap Library
How to write a Bitmap Library

- Look at the spec
How to write a Bitmap Library

- Look at the spec
- Look at some C code
How to write a Bitmap Library

- Look at the spec
- Look at some C code
- The “Aha!” moment
Look at the Specification
Look at the Specification

- Wikipedia
<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Hex Value</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0h</td>
<td>2</td>
<td>42 4D</td>
<td>&quot;BM&quot;</td>
<td>ID field (42h, 4Dh)</td>
</tr>
<tr>
<td>2h</td>
<td>4</td>
<td>46 00 00 00</td>
<td>70 bytes (54+16)</td>
<td>Size of the BMP file</td>
</tr>
<tr>
<td>6h</td>
<td>2</td>
<td>00 00</td>
<td>Unused</td>
<td>Application specific</td>
</tr>
<tr>
<td>8h</td>
<td>2</td>
<td>00 00</td>
<td>Unused</td>
<td>Application specific</td>
</tr>
<tr>
<td>Ah</td>
<td>4</td>
<td>36 00 00 00</td>
<td>54 bytes (14+40)</td>
<td>Offset where the pixel array (bitmap data) can be found</td>
</tr>
</tbody>
</table>

**DIB Header**

<table>
<thead>
<tr>
<th>Offset</th>
<th>Size</th>
<th>Hex Value</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eh</td>
<td>4</td>
<td>28 00 00 00</td>
<td>40 bytes</td>
<td>Number of bytes in the DIB header (from this point)</td>
</tr>
<tr>
<td>12h</td>
<td>4</td>
<td>02 00 00 00</td>
<td>2 pixels (left to right order)</td>
<td>Width of the bitmap in pixels</td>
</tr>
<tr>
<td>16h</td>
<td>4</td>
<td>02 00 00 00</td>
<td>2 pixels (bottom to top order)</td>
<td>Height of the bitmap in pixels. Positive for bottom to top pixel order.</td>
</tr>
<tr>
<td>1Ah</td>
<td>2</td>
<td>01 00</td>
<td>1 plane</td>
<td>Number of color planes being used</td>
</tr>
<tr>
<td>1Ch</td>
<td>2</td>
<td>18 00</td>
<td>24 bits</td>
<td>Number of bits per pixel</td>
</tr>
<tr>
<td>1Eh</td>
<td>4</td>
<td>00 00 00 00</td>
<td>0</td>
<td>BI_RGB, no pixel array compression used</td>
</tr>
<tr>
<td>22h</td>
<td>4</td>
<td>10 00 00 00</td>
<td>16 bytes</td>
<td>Size of the raw bitmap data (including padding)</td>
</tr>
<tr>
<td>26h</td>
<td>4</td>
<td>13 0B 00 00</td>
<td>2835 pixels/meter horizontal</td>
<td>Print resolution of the image.</td>
</tr>
<tr>
<td>2Ah</td>
<td>4</td>
<td>13 0B 00 00</td>
<td>2835 pixels/meter vertical</td>
<td>72 DPI × 39.3701 inches per meter yields 2834.6472</td>
</tr>
<tr>
<td>2Eh</td>
<td>4</td>
<td>00 00 00 00</td>
<td>0 colors</td>
<td>Number of colors in the palette</td>
</tr>
<tr>
<td>32h</td>
<td>4</td>
<td>00 00 00 00</td>
<td>0 important colors</td>
<td>0 means all colors are important</td>
</tr>
</tbody>
</table>
## BMP details

<table>
<thead>
<tr>
<th>APL Representation</th>
<th>Actual Value</th>
<th>Verbal Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>46 0 0 0</td>
<td>70</td>
<td>Size of BMP file</td>
</tr>
<tr>
<td>36 0 0 0</td>
<td>54</td>
<td>Offset to pixel array</td>
</tr>
<tr>
<td>2 0 0 0</td>
<td>2</td>
<td>Image width</td>
</tr>
<tr>
<td>2 0 0 0</td>
<td>2</td>
<td>Image height</td>
</tr>
</tbody>
</table>
Look at C code
typedef struct{
    uint8_t signature[2];
    uint32_t filesize;
    uint32_t reserved;
    uint32_t fileoffset_to_pixelarray;
} fileheader;

typedef struct{
    uint32_t dibheadersize;
    uint32_t width;
    uint32_t height;
    uint16_t planes;
    uint16_t bitsperpixel;
    uint32_t compression;
    uint32_t imagesize;
    uint32_t ypixelspermeter;
    uint32_t xpixelspermeter;
    uint32_t numcolorspallette;
    uint32_t mostimpcolor;
} bitmapinfoheader;
The “Aha!” Moment

- 32 bits are four 8-bit bytes
The “Aha!” Moment

\[ 256 \perp \phi (46 \ 0 \ 0 \ 0) (36 \ 0 \ 0 \ 0) (2 \ 0 \ 0 \ 0) (2 \ 0 \ 0 \ 0) \]

46 36 2 2
How do you represent an image in APL?
How do you represent an image in APL?

- A 2D Matrix of vectors
How do you represent an image in APL?

- A 2D Matrix of vectors
  - Easier to look at while coding
How do you represent an image in APL?

- A 2D Matrix of vectors
  - Easier to look at while coding
  - Slow
How do you represent an image in APL?

- A 2D Matrix of vectors
  - Easier to look at while coding
  - Slow

```apl
content←offset↓bmp
partition←bytes_perpixel{w⊂≃∈(a)\(a÷≃ρw)}contentbitmap←h w ρ partitioned
```
How do you represent an image in APL?

- A 3D Matrix
How do you represent an image in APL?

- A 3D Matrix
  - Faster code with less operations
How do you represent an image in APL?

- A 3D Matrix
  - Faster code with less operations
  - Extensible
How do you represent an image in APL?

- A 3D Matrix
  - Faster code with less operations
  - Extensible

bitmap←⍉width height dimension ρ content
APL and The Web
Passing web-form data with APL

- What is a web-form
Passing web-form data with APL

● What is a web-form
  ○ Syntax is HTML
Passing web-form data with APL

- What is a web-form
  - Syntax is HTML
  - HTTP methods:
    - GET
    - POST
Examples of GET and POST

<form action="code.apl" method="get">
  <input type="submit" value="Click">
</form>

<form action="code.apl" enctype="multipart/form-data" method="post">
  <input type="file" name="name">
  <input type="submit" value="Click">
</form>
Passing form data with APL

- Language agnostic protocol
Passing form data with APL

- Language agnostic protocol
  - GET
    - “&” is the separator
    - environment variable called “QUERY_STRING”
Passing form data with APL

- Language agnostic protocol
  - POST
    - CRLF is the separator (Unicode 13 10)
    - Passed to stdin

```
Contents←{⍵, FIO∆fread 0}⍣{⍺⊢FIO∆feof 0}"'
```
Demo

- Parsing a GET
Demo

- Parsing a GET

\[
\text{QUERY\_STRING} \leftarrow \text{'var1=value1&var2=value2&var3=value3'}
\]

\[
R \leftarrow ',\text{QUERY\_STRING}
\]

\[
(+R='\&') \subset R
\]
Demo

- Parsing a GET

```
QUERY_STRING ← 'var1=value1&var2=value2&var3=value3'
R ← '&', QUERY_STRING
(+\R='&') ⊂ R
```

```
&var1=value1 &var2=value2 &var3=value3
```
Other Goodies
Indexing
Indexing

- Good algorithms are Index-Origin independent
Indexing

- Good algorithms are Index-Origin independent
- How to convert from \( \text{IO} \leftarrow 0 \) to \( \text{IO} \leftarrow 1 \)

\[
\begin{align*}
\text{ALEX}[2 3 4] & \quad \circ \quad \text{IO} \leftarrow 1 \\
\text{ALEX}[1 2 3] & \quad \circ \quad \text{IO} \leftarrow 0 \\
\text{ALEX}[\text{IO} + 1 2 3] & \quad \circ \quad \text{Doesn’t matter}
\end{align*}
\]
Control structures

- None are built in
Control structures

- None are built in
- Branching and line labels are included.
Control structures

LDI r16,0b00000001
CPI r16,0b00000001
BRNE somewhere

RJMP continue
somewhere:
continue:

r16←1
result←r16=1
→(result≠0)/somewhere

→continue
somewhere:
continue:
Questions?
Thank You!