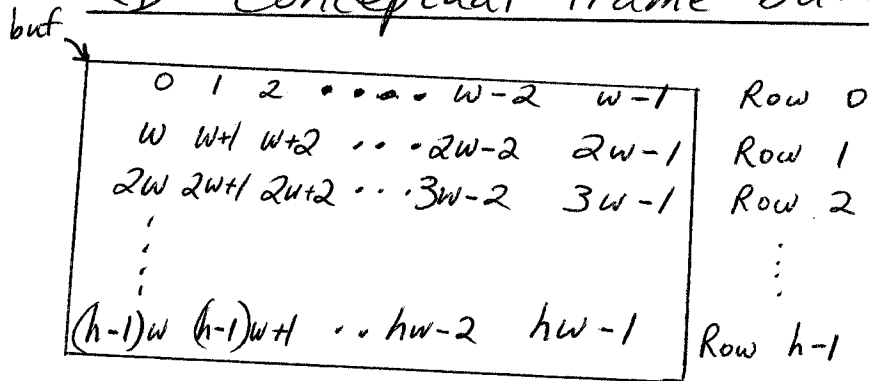


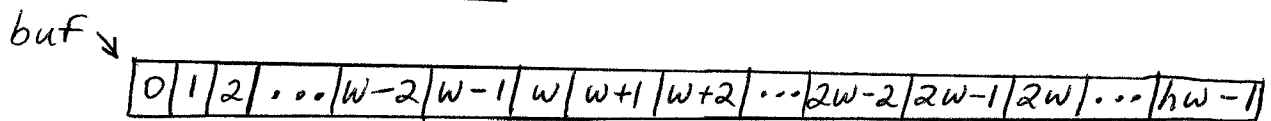


## 2D conceptual frame buffer organization:



Address of point  $(x, y)$  is  $\text{buf}[y][x]$   
This is identical to  $\text{buf} + y * w + x$

## 1D representation:



One row follows another in this linear array

Address of point  $(x, y)$  is  $\text{buf} + y * w + x$

When visiting successive pixels along a row it is sufficient to increment pointer (e.g.  $P++$ )

`writePixel(x, y, val)`

`int x, y;`

`unsigned char val;`

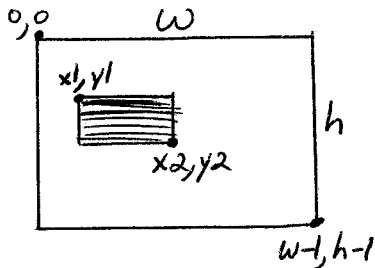
`{`

`buf[y * FrameBufferWidth + x] = val;`

`}`

↑ ↑  
global variables

## Example:



Fill sub-rectangle with 100:

```
val = 100;
```

```
for(y=y1; y<y2; y++)
```

```
  for(x=x1; x<x2; x++)
```

```
    writefixel(x, y, val);
```



Inefficient because writefixel is most useful for random access. However, successive pixels along a row can be accessed sequentially.

Better solution:

```
val = 100;
```

```
for(y=y1; y<y2; y++) {
```

```
  p = buf + y * FrameBufferWidth + x1;
```

```
  for(x=x1; x<x2; x++)
```

```
    *p++ = val;
```

```
}
```

↑  
sequential access is more efficient