

Memory Management

Virtual Memory



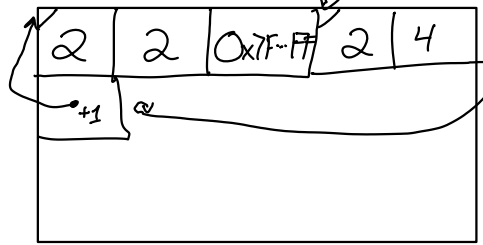
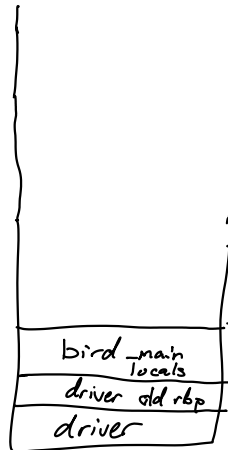
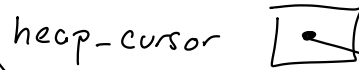
```

def f n =
  if n > 0 then
    let x = f (n-1) in
      (n, x)
  else
    false
end
f 2
    
```

Registers



Data Section



Stack

Heap

(2, (1, false))

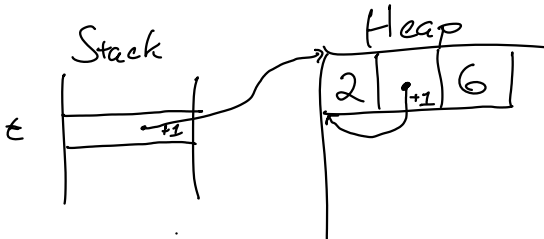
Gull

```

<expr> ::= ...
| <expr> [<expr>] := <expr>
    
```

```

let t = (2, 3) in
let s = t[0] := t in
t
    
```

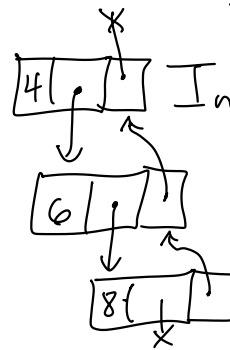


```

let t = (2, 3) in
let s = t[0] := 4 in
(s, t)
    
```

⇓
(4, (4, 3))

1. update pos 0 to be 4
2. evaluate to 4



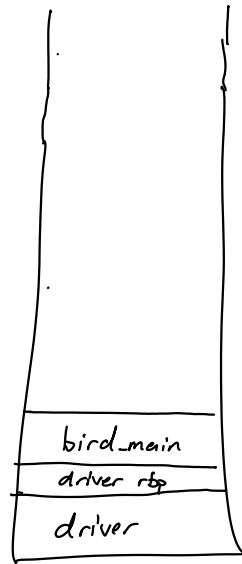
```

C:
int x;
int y;
x = (y = 0);
    
```

```

def f n =
  let x = (n, n) in
  if n = 0 then
    x
  else
    f (n-1)
end
f 2

```



Stack

RAX +1

2	4	4	2
2	2	2	0
0			

Heap

"Garbage" - memory which is allocated but no longer used

- ① Require programmer to acknowledge when they don't need memory
- ② Do it for them