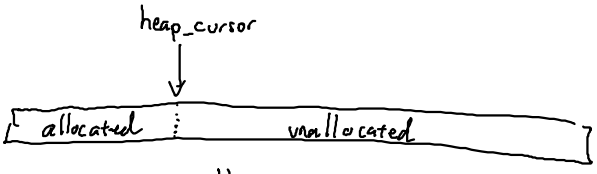
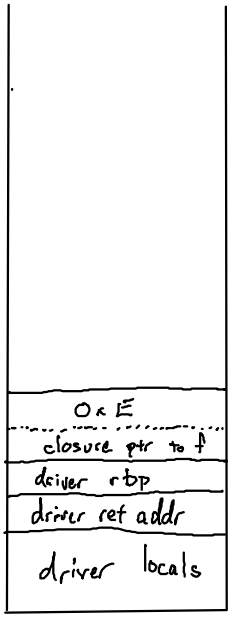
Eagle

1. Allocate heap memory
2. Use heap_cursor to track usage
3. If heap_cursor reaches the end of heap space, !!



```
def f n =
    if n > 8 then
        n
    else
        (n, f(n+1))
end
f 7
```



heap_cursor	0x7030
rax	0x7001

0x7000

0x2	0xE	0x7019	0x2
0x10	0x12		

(7, (8, (9)))

Gull

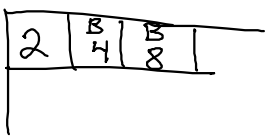
1. Syntax

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

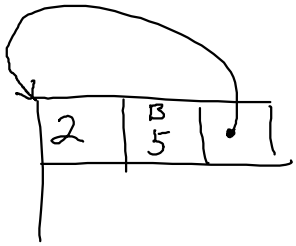
2. Semantics (example)

```
let t = (5,8) in
let j = (t[0] := 4) in
t[0] + t[1] + j
```

4 + 8 + 4 = 16




```
let t = (5,8) in
let j = (t[1] := t) in
(* t[1][1][1][1][1][0] *)
t
```



```
def f n =
  let j = print ((n, n)) in 4
end
def g n =
  if n < 5 then
    n
  else
    f n + g (n-1)
end
g 6
```