

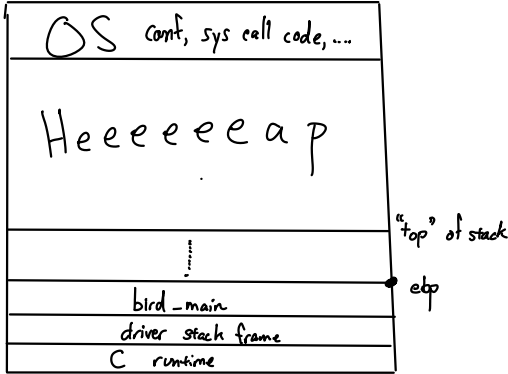
$\langle \text{expr} \rangle ::= \dots$
 | first($\langle \text{expr} \rangle$)
 | second($\langle \text{expr} \rangle$)
 | ($\langle \text{expr} \rangle, \langle \text{expr} \rangle$)
 | ispair($\langle \text{expr} \rangle$)

first((4,5)) \Rightarrow 4

let x = (4,5) in second(x) \Rightarrow 5

(1+2, 4+3) \Rightarrow (3, 7)

first(let x = (2, false) in x) \Rightarrow 2



Allocating Memory

syscall : brk

malloc

Our Heap

1. Driver calls malloc
2. Passes ptr to "heap" into bird-main
3. As program runs, use memory at ptr as heap allocation

What we're NOT doing

1. Allocating exactly enough heap memory
2. Checking heap overflow
3. Reclaiming memory

How to Describe Heap Things?

Binary repr:

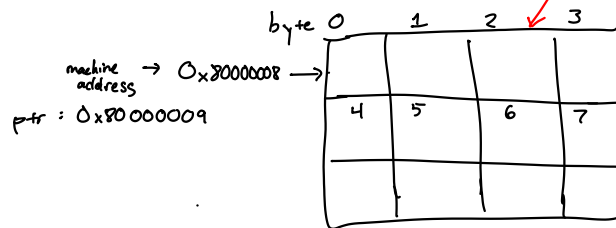
Machine Values

Bird Values

booleans: $0x[b111]FFFFFF \Rightarrow b$ (true=1, false=0)

ints: $0xNNNNNNNN[nnn0] \Rightarrow 0xNNNNNNNNnnn$ as int

ptrs: $0xNNNNNNNN[nn01] \Rightarrow 0xNNNNNNNN[nn00]$ as ptr



type expr = ...
 | EPair of expr * expr

⋮
 | EPair(e1, e2) →
 ce env e1
 (let env' = alloc-local env in
 ... e2
 ... env')

1. Put my two results in heap at [heap-cursor] and [heap-cursor + 4]
2. Store current heap cursor in a register (eax)
3. Increase heap cursor by 8
4. or eax, 1

section .data ← data section
 align 4
 heap-cursor:
 dd 0

Section .text ← code section
 bird-main:
 ← copy 1st param into [heap-cursor]

loads the value of the heap cursor →
 (not the first word of the heap)

mov eax, [heap-cursor]

get 1st word of heap →

mov ebx, [eax]

(3, 8)

