ECE 350
Lab Tutorial 1

Aravind Vellora Vayalapra

WHAT TO EXPECT FROM THESE TUTORIALS.

- We will have three – 1 before each project is due
- 9 AM

WHAT TO EXPECT FROM THESE TUTORIALS.

- Overview of the labs
- Present a debug “how-to”
- Not an in-depth review – lab manual
- Ask me questions

Lab manual is on the course website!

LAB GROUPS

- Groups of 4. We may okay 3.
- Not 5. Not 2.
- Group policy in the lab manual.

LAB MACHINES

- Remote Desktop to Lab machines
- Instructions in the lab manual
- ece-mcu* machines

GIT

- Each group will have a Git repository
- Starter code will be given to you
- SSH keys on ECE lab machines

LAB

- Develop a real-time kernel
- P1 – Memory management.
- P2 – Task management
- P3 – Inter task communication and IO

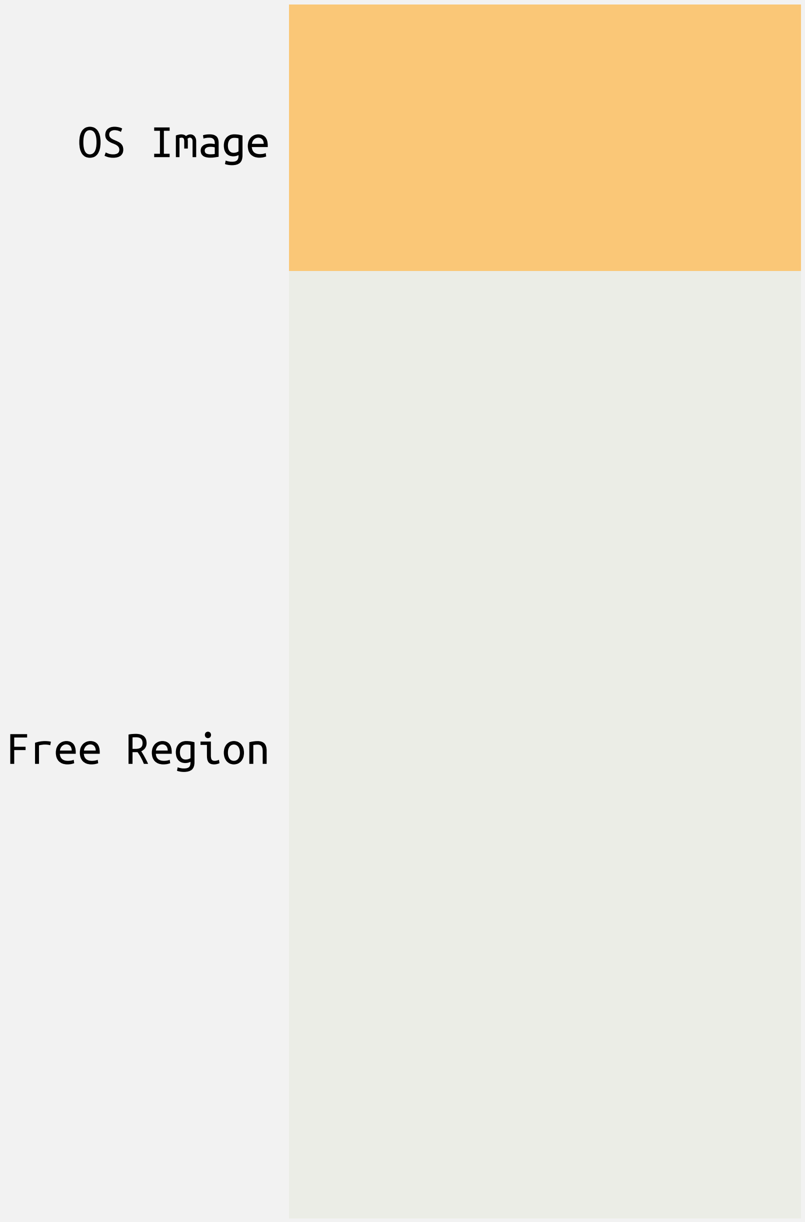
LAB

- ARM A9 Core
- Terasic DE1-SoC
- C and ARM Assembly
- ARM DS IDE for debugging

P1: OVERVIEW

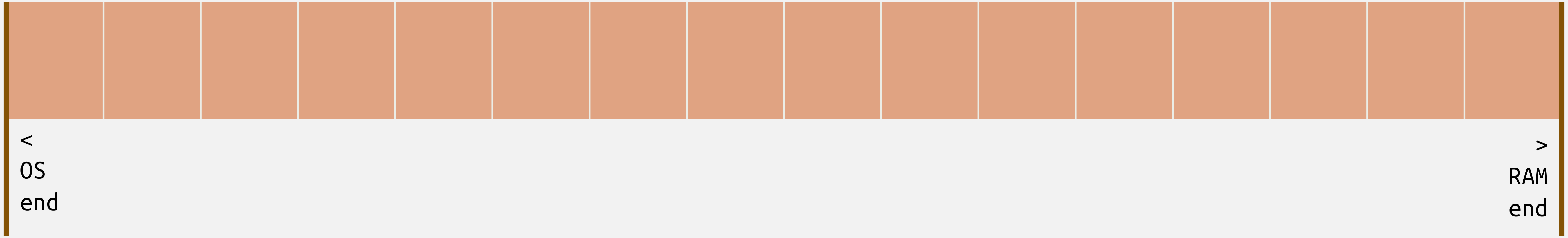
- Memory management
- Develop your own malloc
- Testing, debugging

P1:OVERVIEW

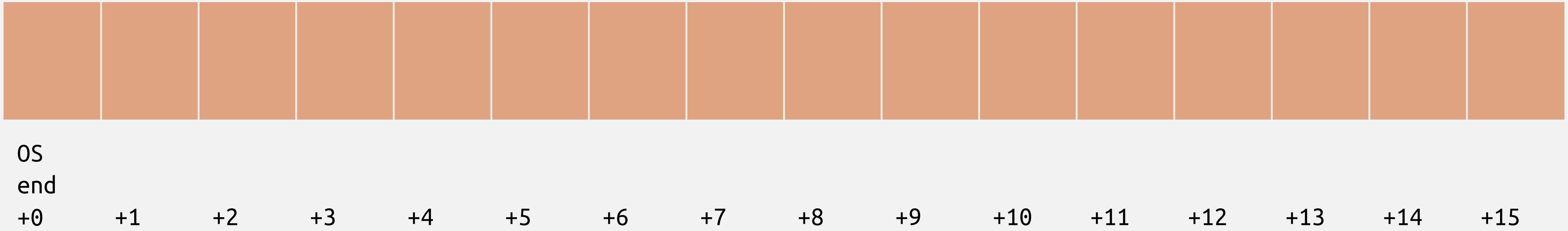


P1:MEMORY MANAGEMENT

- Manage the Free Region using first-fit
- Implement memory management API
    - mem_init
    - mem_alloc
    - mem_dealloc


< OS end
> RAM end

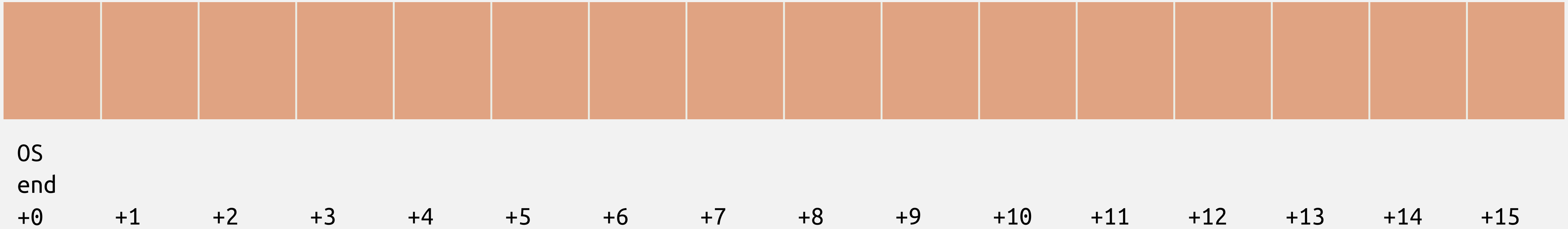
- Assume here each block is 4 bytes
- We have 16 x 4 bytes in total
- We need to manage this memory



OS
end
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +10 +11 +12 +13 +14 +15

T0: request 4 bytes

`mem_alloc(4)`

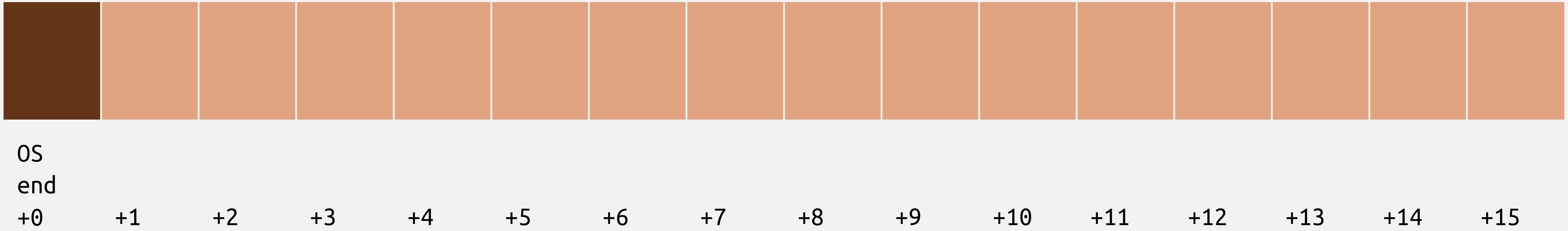


OS
end

+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +10 +11 +12 +13 +14 +15

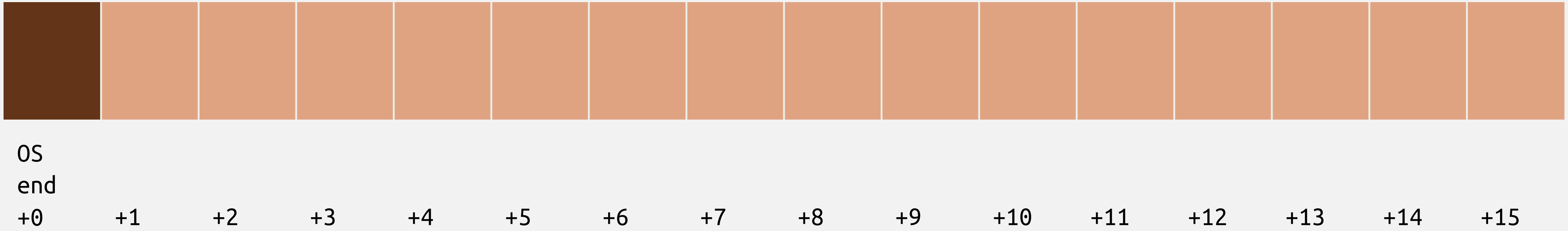
T0: allocate 4 bytes

```
return OS_END + 0
```



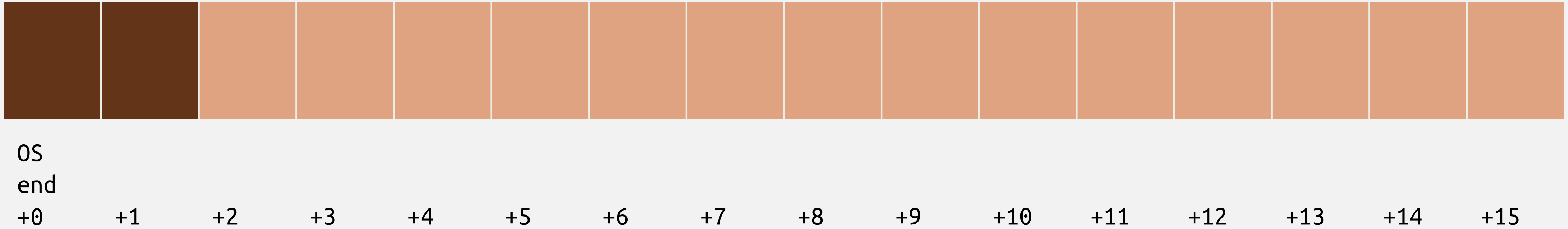
T1: request 4 bytes

```
mem_alloc(4)
```



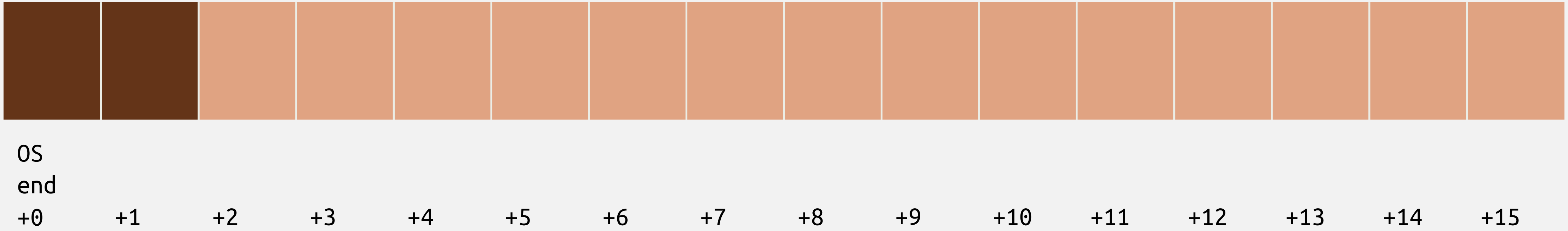
T1: allocate 4 bytes

```
return OS_END + 1
```



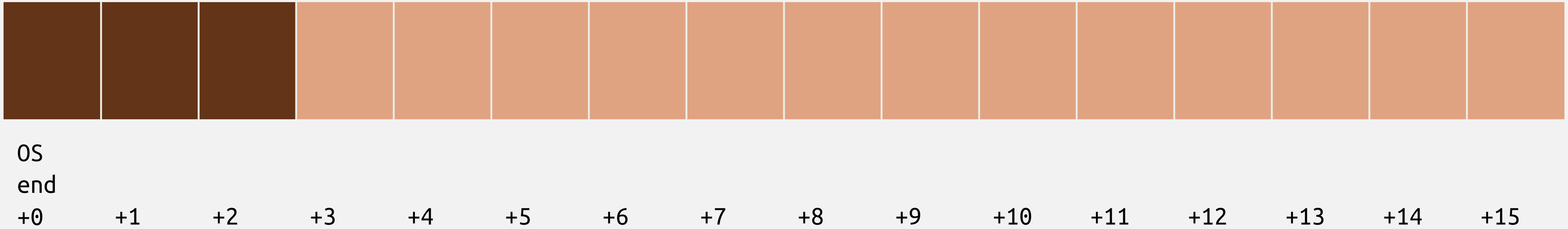
T2: request 4 bytes

`mem_alloc(4)`



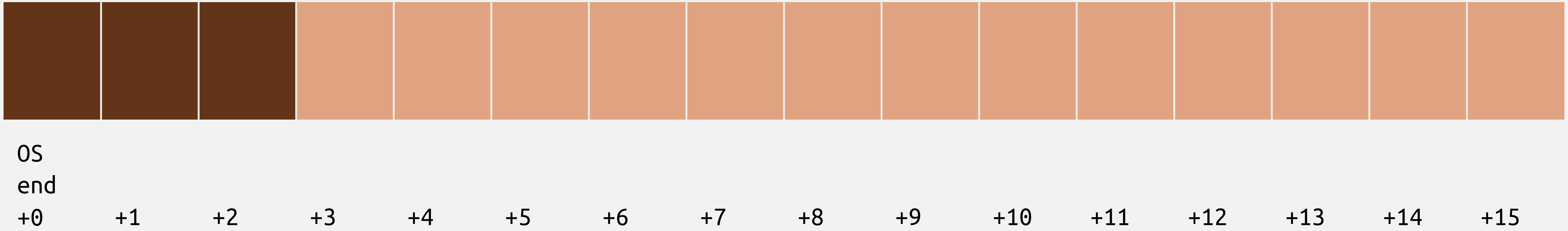
T2: allocate 4 bytes

```
return OS_END + 2
```



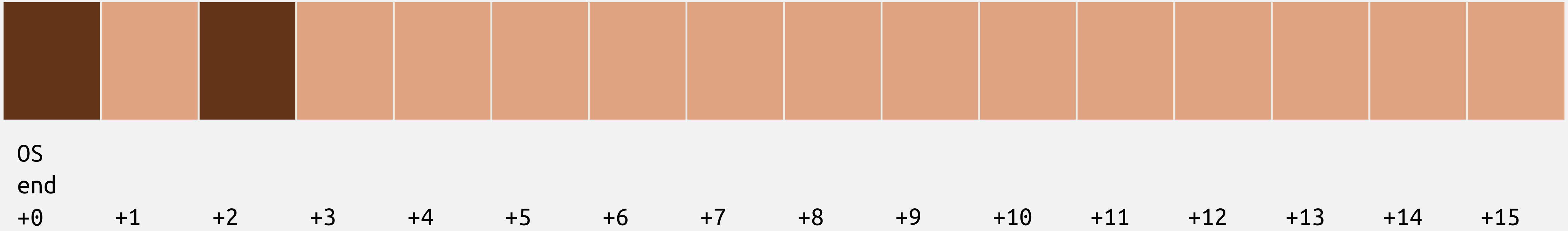
T3: deallocate OS_END + 1

```
mem_dealloc(os_end + 1)
```



T3: deallocate OS_END + 1

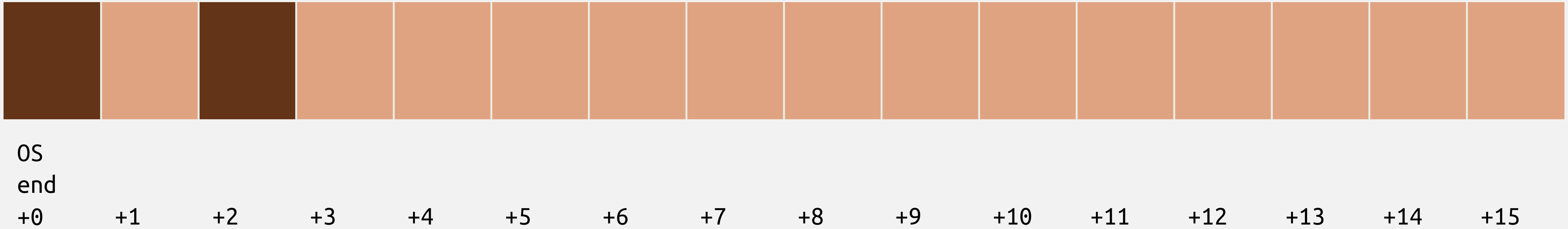
```
return RTX_OK
```



OS
end
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +10 +11 +12 +13 +14 +15

T4: request 8 bytes

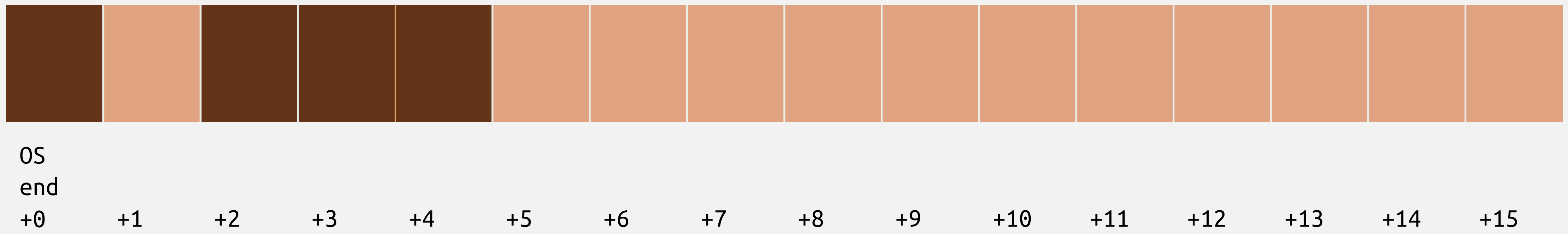
`mem_alloc(8)`



OS
end
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9 +10 +11 +12 +13 +14 +15

T4: allocate 8 bytes

```
return OS_END + 3
```



P1:MEMORY MANAGEMENT

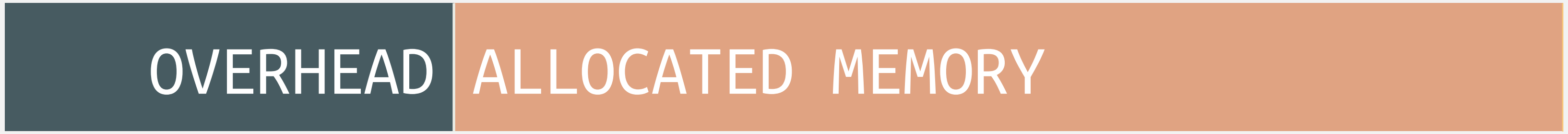
- Keep track of allocated memory
- Allocation are aligned to 4 bytes
- Return the memory spot that is “first-fit”

P1:IMPLEMENTATION

- Overhead to keep track of memory
- Linked list

P1:IMPLEMENTATION

- Overhead to keep track of memory
- Linked list
- Overhead at beginning



OVERHEAD ALLOCATED MEMORY

P1:METRICS

- Throughput; Requests per time
  - Use the A9 private timer to measure request time
- Heap utilization ratio; how much heap is used
  - Allocate close to the entire heap and evaluate overhead
  - Minimize overhead!

WALK THROUGH ARM DS IDE

QUESTIONS?