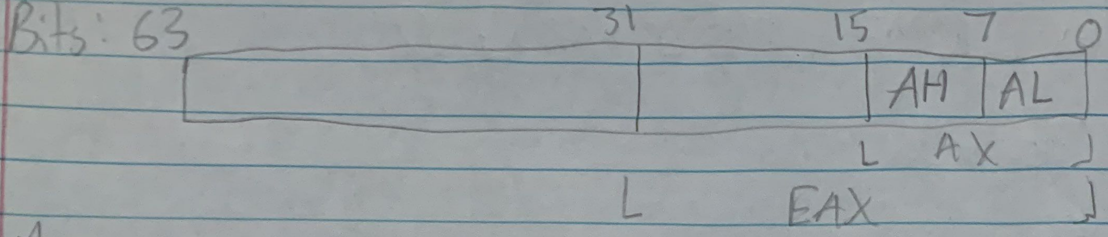

Memory

Highest 0xFFFF...

Kernel mode

...

Stack (grows down)

...

Memory mapping

File mapping

Anon mapping

...

Heap

...

BSS Seg

Data Seg

Text Seg (start of program)

0x00000...

- rax holds syscall num
- args 1-6 in rdi, rsi, rdx, r10, r8, r9
- return value put in rax
- rbx, r12-15 preserved

rax = rax(rdi, rsi, rdx, r10, r8, r9)

NASM syntax

label: instruction operands ; comment

```
INSTR
INSTR arg
INSTR dst, src
INSTR dst, src, aux
```

Immediate value storage

```
MOV eax, 0x000F
```

Sizes

name	BYTE	WORD	DWORD	QWORD	TWORD	YWORD	ZWORD
bytes	1	2	4	8	16	32	64

Ex CMP BYTE [rax+rbx], 0

Operands:

R- register

I- immediate

0x	hex
0o	oct
0b	bin
''	char
""	str

M -memory (effective addr.)

[label]	data at label
[label+1]	data at label + const offset
[label+register]	label offset by amt in reg

mov al, [esi + ecx*8+100] ⇒ al=array[ecx*8+100]

Labels refer to address of their line

-starting with . means local addr.

```
func1: ...
.loop: ...  <

func2: ...
.loop: ...
     jmp func1.loop
```

SECTION .bss, .data, .text

GLOBAL - exports symbol

EXTERN - imports

RESB, RESW, RESD, RESQ

x: equ 10

define x as 10 instead of address