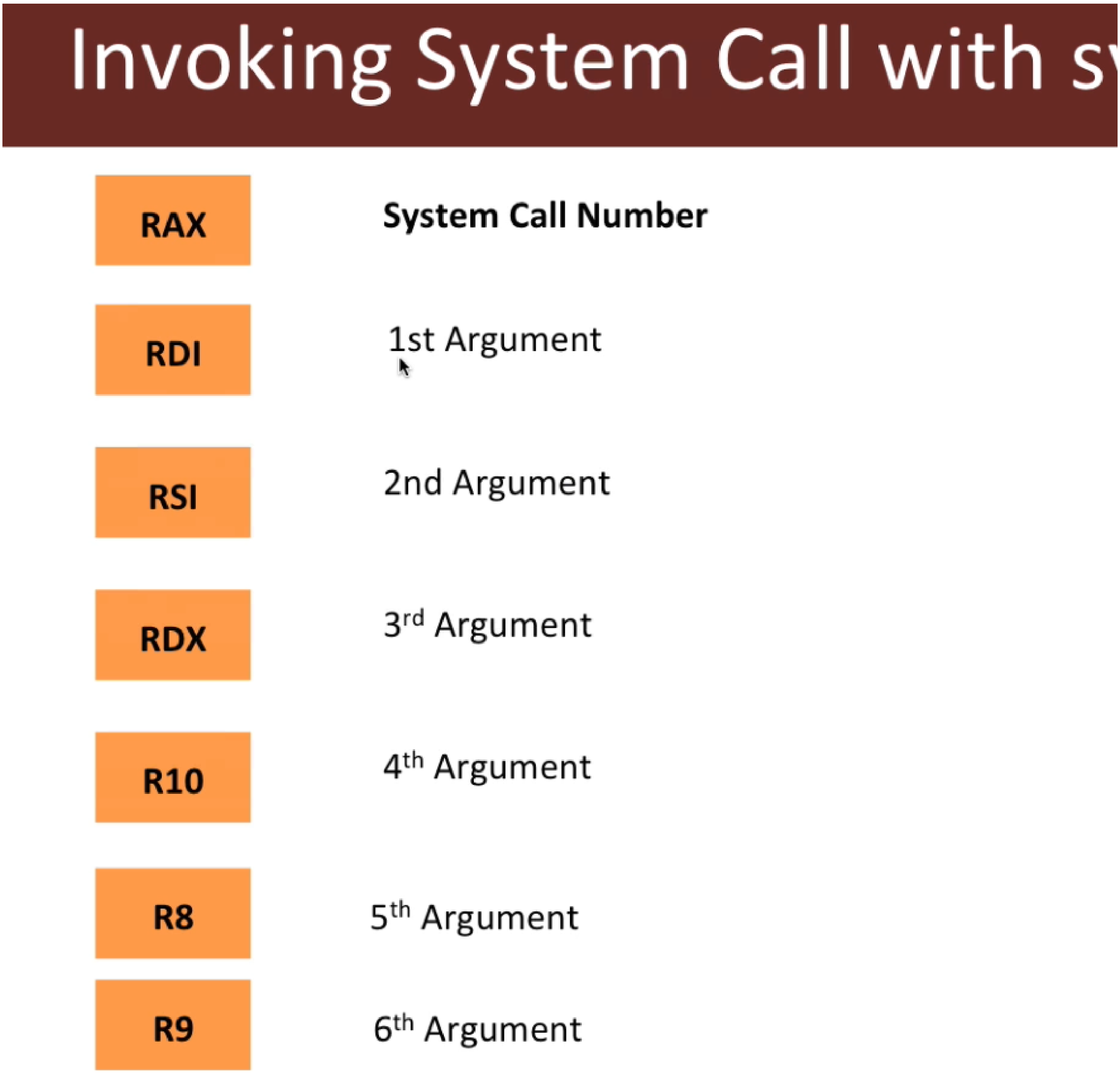
Hello World

We have a option to choose syntax (AT&T or Intel flavor)

- `vim helloworld.nasm`
- We have to declare text section as
  - section .text
- And data section also.
  - section .data
    - `hello_world : db 'Hello world'`
    - Db means **DataByte**
- in Ubuntu all System calls defined in
  - `vim /usr/include/x86_64-linux-gnu/asm/unistd_64.h`

To use syscall we have to use registers :



Invoking System Call with s

RAX	**System Call Number**
RDI	1st Argument
RSI	2nd Argument
RDX	3rd Argument
R10	4th Argument
R8	5th Argument
R9	6th Argument

```
section .text


_start:

    ;print screen


    mov rax ,1

    mov rdi, 1

    mov rsi, hello_world
    mov rdx, length
    syscall

    ; exit gracefully

    mov rax, 60
    mov rdi, 11
    syscall


section .data
    hello_world: db 'Hello world vishal from world'
    length : equ $-hello_world
```

- this is the code
- `nasm -f elf64 hello.nasm -o hello.o`
- `ld hello.o -o hello`

Then you can run it.

- Efficient Code :

```
section .text

_start:
    ;print screen
    xor rax,rax
    mov al,1
    mov dil, 1
    mov rsi, hello_world
    mov dl, length
    syscall

    ; exit gracefully

    mov al, 60
    mov dil, 11
    syscall


section .data
    hello_world: db 'Hello world vishal from world'
    length : equ $-hello_world
```