Home.html

Assembly.s

```
Assembly ‘Language’ {
    [For Beginners Workshop]

        < Here is where your presentation begins >

}
```

Step < /1 > {

< Compilation, architectures processeur, mémoire, assembleur >

}

Step < /2 > {

< Hello_World.asm >

}

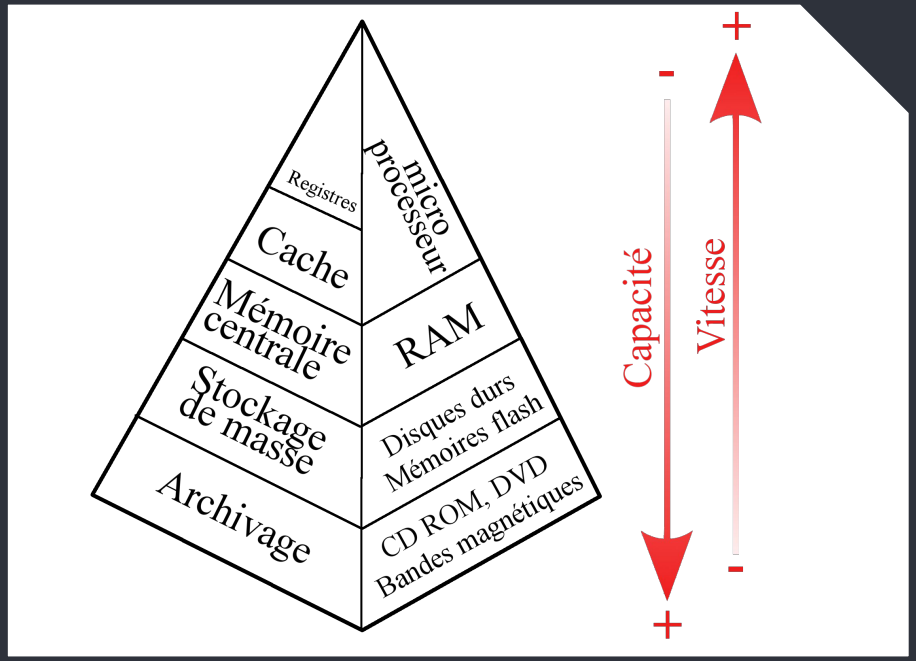
```
01 {

    [Step 0×01]

        < Compilation, architectures
        processeur, mémoire, assembleur >

}
```

Types de ‘Mémoires’;



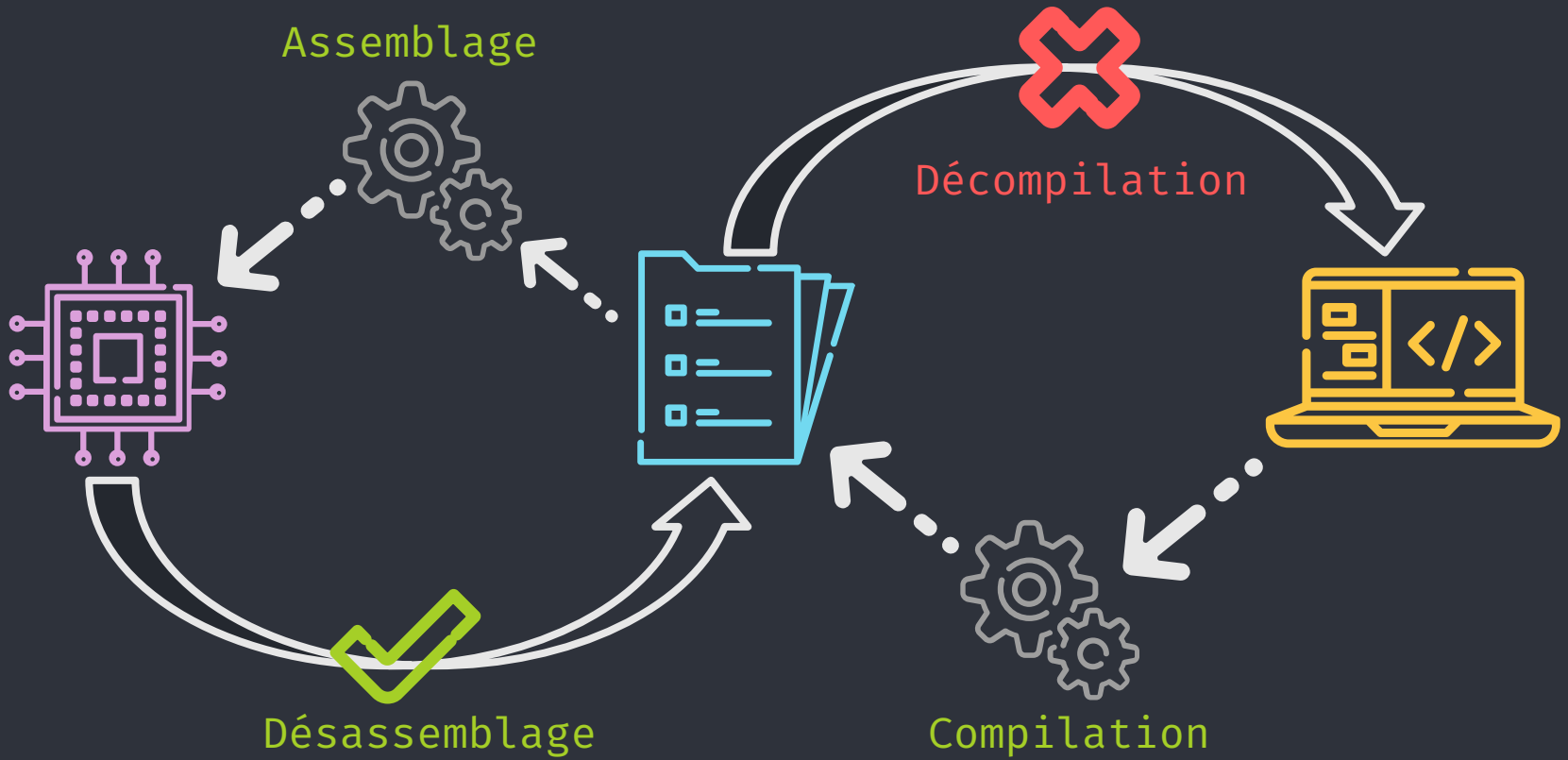
Architectures ‘Processeur’;

RISC

- ARM
  - ARM 32 bits
  - ARM64
- MIPS

CISC

- Intel
  - x86 (i386)
  - x86_64 (amd64)


Assemblage
Décompilation
Désassemblage
Compilation

Language ‘Assembleur’;

$42_{10} \iff 0\times2A \iff 52_{8} \iff 0010\ 1010_{2}$

```
48 31 ff                ⟺    xor    %rdi,  %rdi
48 c7 c0 3c 00 00 00    ⟺    mov    $0×3c, %rax
0f 05                   ⟺    syscall
```

Syntaxe {

- INTEL

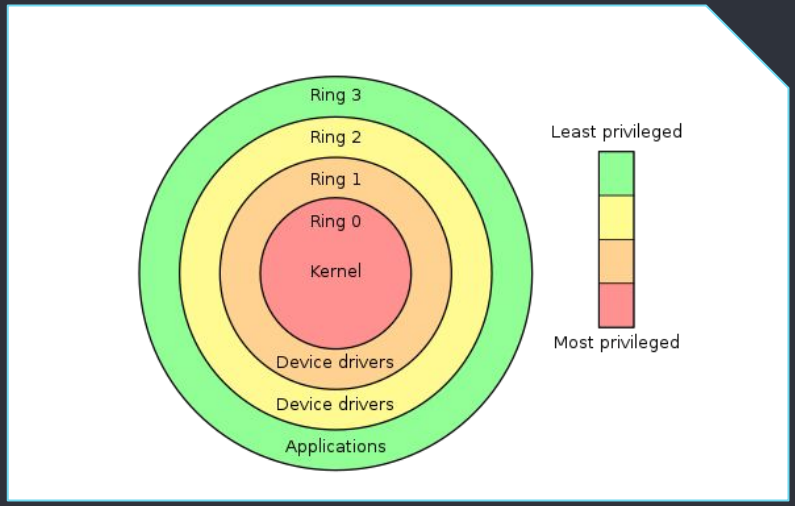
```
mov out, in
1: mov al, 1
2: mov DWORD PTR[-4 +ebp], eax
```

- AT&T

```
mov in, out
1: movb $1, %al
2: mov %eax, -4(%ebp)
```

}

```
1  Introduction; {
2
3
4          Mov
5          Lea
6          Inc / dec
7          Push / pop
8          Add / sub
9          Syscall
10         Jump / je / jne / …
11         Call / ret
12
13 }
14
```



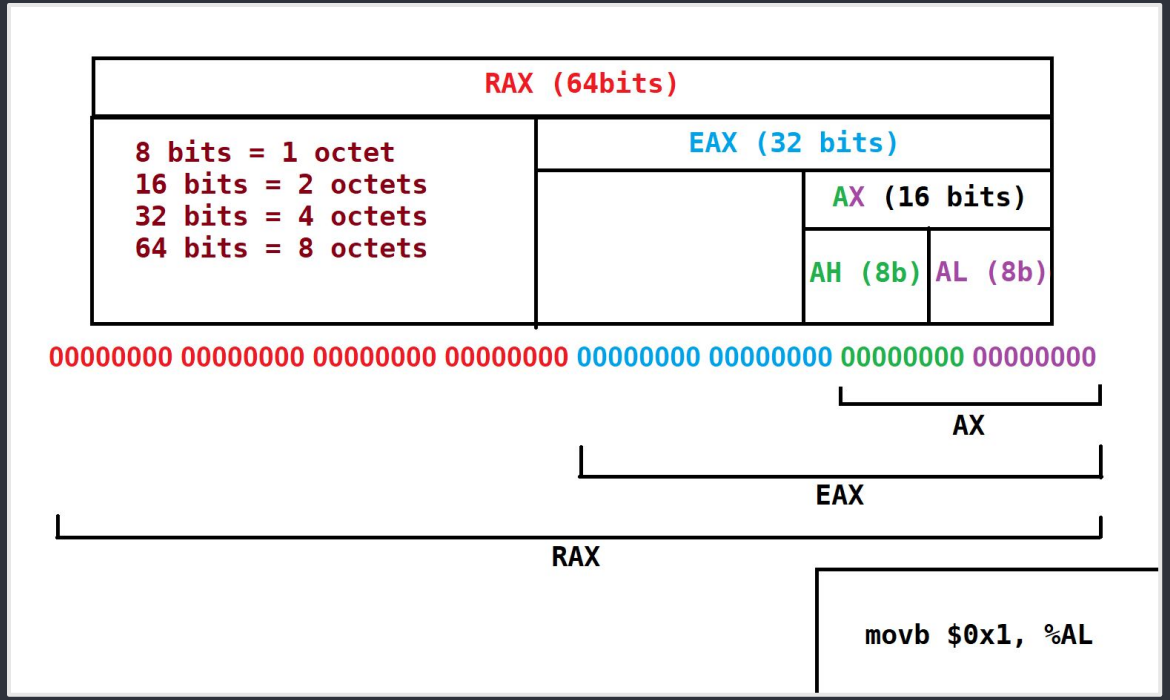
Registres{

Généraux	Pointeurs	Segments
AX	SP	CS
BX	BP	DS
CX	SI	SS
DX	DI	ES

IP	FLAGS

}

Registres{


RAX (64bits)
8 bits = 1 octet
16 bits = 2 octets
32 bits = 4 octets
64 bits = 8 octets
EAX (32 bits)
AX (16 bits)
AH (8b)
AL (8b)
00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
AX
EAX
RAX
movb $0x1, %AL

}