ARCHITECTURE DES ORDINATEURS

11)- Introduction à l' ASSEMBLEUR 'x86' *(VI)*

Biblio ::

1/« ***Assembly Langage for INTEL-based computers*** »
[Kip R. IRVINE] – Ed. Prentice Hall, 1999 – ISBN: 0-13-660390-4.

2/« ***An assembly langage introduction to computer architecture*** *(using the intel pentium)*»
[K. MILLER] – Ed. Oxford University Press, 1999 – ISBN: 0-19-512376-X

3/«***The Intel microprocessors : Architecture, Programming & Interfacing***»
[Barry B. BREY] – Ed. Prentice Hall 2006 – ISBN: 0-13-119506-9.

4/ «***The Hardware Bible***»
[Winn L. Rosch] – Ed. QUE /McMillan computer Publishing – ISBN: 0-7897-1743-3.

@ web::

1-http://css.csail.mit.edu/6.858/2013/readings/
2-http://www.ustudy.in/node/

11)- Introduction à l’ASM ‘x86’ *(VI)*

```
; exemple / illustration = ‘Hello World ‘
title hello World Program            (hello.asm)

.model small
.stack 100h
.data
message db “hello, world !”,0dh,0ah, ‘$’

.code
main proc
            mov ax, @data
            mov ds, ax

            mov ah, 9
            mov dx, offset message
            int 21h

            mov ax, 4C00h
            int 21h
main endp
end main
```

NB: **‘29 Ko’** en **ASM** *vs* **‘562 Ko’** en **C** !!

11)- Introduction à l’ASM ‘x86’ *(VI)*

```
; exemple / illustration = ‘Hello World ‘
title hello World Program                    (hell [illegible] m)

.model small
.stack 100h
.data
message db “hello, world !”,0dh,0ah, ‘$’

.code
main proc
        mov ax, @data
        mov ds, ax

        mov ah, 9
        mov dx, offset message
        int 21h

        mov ax, 4C00h
        int 21h
main endp
end main
```

« directive » :: besoin
Mémoire_prg =‘64 Ko’

11)- Introduction à l’ASM ‘x86’ *(VI)*

```
; exemple / illustration = ‘Hello World ‘
title hello World Program                    (hello.asm)

.model small
.stack 100h
.data
message db “hello, world !”,0dh,0ah, ‘$’

.code
main proc
        mov ax, @data
        mov ds, ax

        mov ah, 9
        mov dx, offset message
        int 21h

        mov ax, 4C00h
        int 21h
main endp
end main
```

« ***directive*** » :: Réservation
mémoire_PILE ‘256 (o)’

11)- Introduction à l’ASM ‘x86’ *(VI)*

```
; exemple / illustration = ‘Hello World ‘
title hello World Program                         (hello.asm)

.model small
.stack 100h
.data
message db “hello, world !”,0dh,0ah, ‘$’

.code
main proc
            mov ax, @data
            mov ds, ax

            mov ah, 9
            mov dx, offset message
            int 21h

            mov ax, 4C00h
            int 21h
main endp
end main
```

« directive » :: Déclaration espace_**mémoire (DATA)**

« Déclaration » ::

- Variable ‘**message**’
- **‘db’** : Représentation **‘8 bits’** en mémoire

7)- Introduction à l’ASM ‘x86’ *(II)*

```
; exemple / illustration = ‘Hello World ‘
title hello World Program                    (hello.asm)

.model small
.stack 100h
.data
message db “hello, world !”,0dh,0ah, ‘$’

.code
main proc
        mov ax, @data
        mov ds, ax

        mov ah, 9
        mov dx, offset message
        int 21h

        mov ax, 4C00h
        int 21h
main endp
end main
```

« ***directive*** » :: Déclaration (début) **Code_Segment**

11)- Introduction à l’ASM ‘x86’ *(VI)*

```
; exemple / illustration = ‘Hello World ‘
title hello World Program                    (hello.asm)

.model small
.stack 100h
.data
message db “hello, world !”,0dh,0ah, ‘$’

.code
main proc
        mov ax, @data
        mov ds, ax

        mov ah, 9
        mov dx, offset message
        int 21h

        mov ax, 4C00h
        int 21h
main endp
end main
```

« ***déclaration*** » :: Début & fin
« **Main** »

11)- Introduction à l’ASM ‘x86’ *(VI)*

```
; exemple / illustration = ‘Hello World ‘
title hello World Program                              (hello.asm)

.model small
.stack 100h
.data
message db “hello, world !”,0dh,0ah, ‘$’

.code
main proc
          mov ax, @data
          mov ds, ax

          mov ah, 9
          mov dx, offset message
          int 21h

          mov ax, 4C00h
          int 21h
main endp
end main
```

Association :: DS <-> @data

11)- Introduction à l'ASM 'x86' *(VI)*

```
; exemple / illustration = 'Hello World '
title hello World Program                    (hello.asm)

.model small
.stack 100h
.data
message db "hello, world !",0dh,0ah, '$'

.code
main proc
          mov ax, @data
          mov ds, ax

          mov ah, 9
          mov dx, offset message
          int 21h

          mov ax, 4C00h
          int 21h
main endp
end main
```

Action ::

- (ah=9) & (int 21h) ⇔ affichage moniteur
- (dx) :: @ E/S (moniteur)

11)- Introduction à l'ASM 'x86' *(VI)*

```
; exemple / illustration = 'Hello World '
title hello World Program                         (hello.asm)

.model small
.stack 100h
.data
message db "hello, world !",0dh,0ah, '$'

.code
main proc
        mov ax, @data
        mov ds, ax

        mov ah, 9
        mov dx, offset message
        int 21h

        mov ax, 4C00h
        int 21h
main endp
end main
```

Action ::

- (ax=4C00h) & (int 21h) ⇔ fin_prg & retour au S.E.