ARCHITECTURE DES ORDINATEURS

(10)- INTRODUCTION à l' ASSEMBLEUR 'x86' *(V)*

LES INTERRUPTIONS

(10)- INTRODUCTION à l' ASSEMBLEUR 'x86' ***(V)***

LES INTERRUPTIONS

ILLUSTRATION :: **RAPPEL**

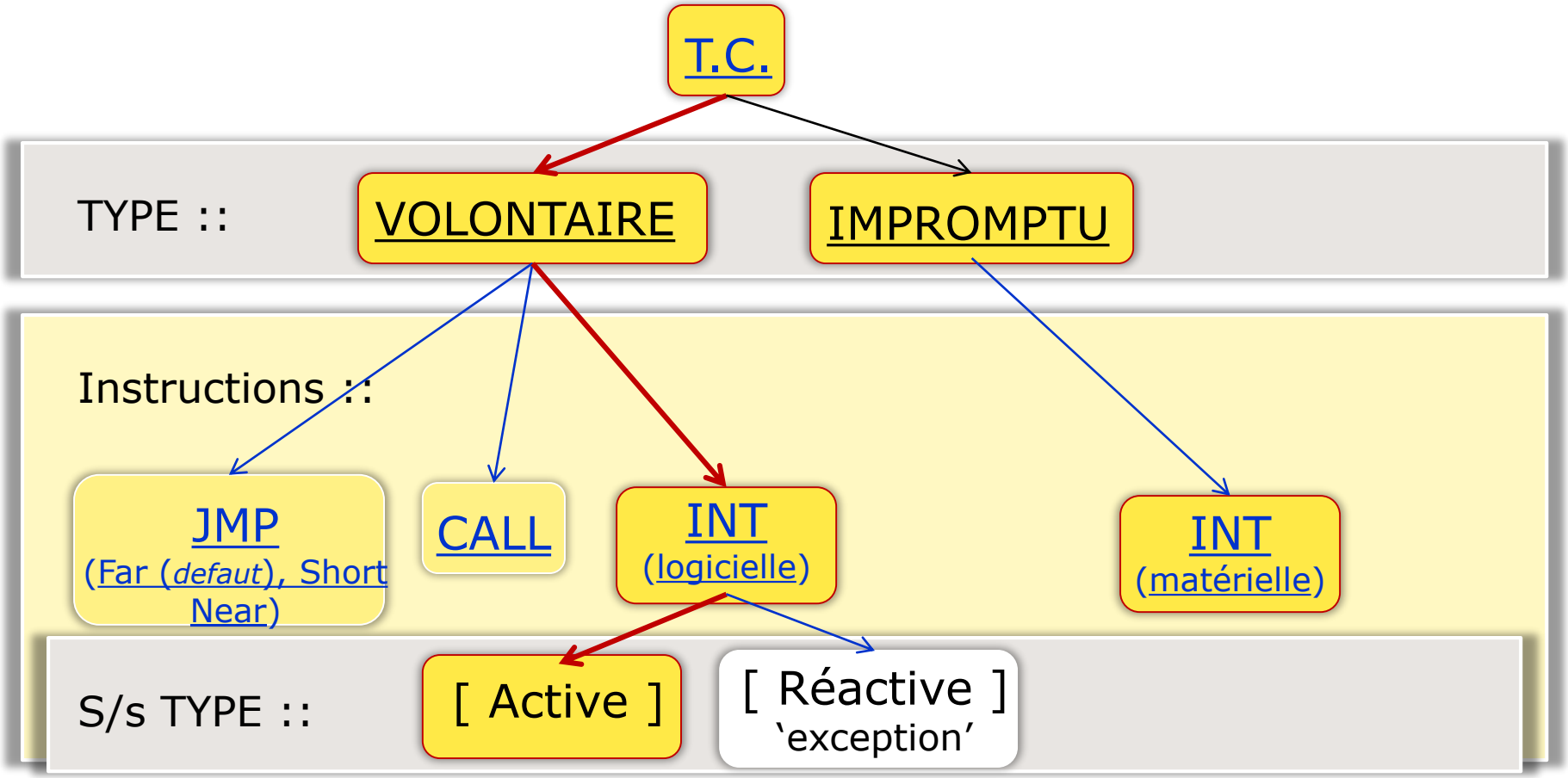
```
; exemple / illustration = 'Hello World '
title hello World Program              (hello.asm)
.model small
.stack 100h
.data
message db "hello, world !",0dh,0ah, '$'
.code
main proc
        mov ax, @data
        mov ds, ax

        mov ah, 9
        mov dx, offset message
        int 21h          ; → appel à la fct DOS "21" pour affichage VIDEO de
                         ; "message"
        mov ax, 4C00h
        int 21h          ; → appel à la fct DOS "21" pour  <sortie= Fin de
                         ; PROGRAMME>
main endp
end main
```

(10)- INTRODUCTION à l' ASSEMBLEUR 'x86' ***(V)***

LES TRANSFERTS de CONTROL :: INTERRUPTIONS

TYPOLOGIE & CLASSIFICATION :: **RAPPEL**



(10)- INTRODUCTION à l' ASSEMBLEUR 'x86' ***(V)***

LES TRANSFERTS de CONTROL :: INTERRUPTIONS

Dans le slide précédent, on rappelle que 'INT' peut être:
1) **'matérielle'**, donc initiée par l'occurrence d'un événement externe au PC, qui viendra 'demander' au CPU d'honorer sa requête (request) en exécutant une fonction spécifique, préalablement programmée et installée en mémoire.
2) **'logicielle'**, donc suivant la volonté du programmeur, qui demande l'exécution d'une fonction précise en spécifiant son numéro et en préparant préalablement les paramètres indispensables au bon déroulement de cette fonction; exple, dans le *slide 2*, on appelle INT 21 avec les paramètres AH=09h, ou AX=4C00h selon le cas.
Cette interruption est dite active car elle suit la volonté instantanée du programmeur; elle est dite réactive lorsqu'un état de fonctionnement du CPU aboutit à une situation 'anormale' nécessitant l'intervention du SE; cette intervention du SE est dite 'exception': exple, tentative d'ecriture hors pile, demande d'opération interdite, . . .etc.
Dans tous ces cas, la fonction dont l'exécution est sollicitée sera dite 'Routine d'Interruption'.
Tous ces cas d'interruptions obéissent à un mécanisme d'exécution qui fait des INT (interruptions) des sous-cas de TC (Transfert de Control).

(10)- INTRODUCTION à l' ASSEMBLEUR 'x86' ***(V)***

LES INTERRUPTIONS

Définition:

Evénement LOGICIEL ou MATERIEL, VOLONTAIRE ou IMPROMPTU, provoquant la « suspension » d'un traitement au profit d'un autre, dit [**R**outine d'**INT**erruption].

(10)- INTRODUCTION à l' ASSEMBLEUR 'x86' ***(V)***

LES INTERRUPTIONS

Définition:

Evénement LOGICIEL/MATERIEL, VOLONTAIRE/IMPROMPTU, provoquant la suspension d'un traitement au profit d'un autre, dit [**R**outine d'**INT**erruption].

Note 1 :
Lorsque la [**R**outine d'**INT**erruption] est prédéfinie au niveau DOS (*resp*^{t}. BIOS), elle est dîte fonction DOS (*resp*^{t}. BIOS)

Note 2 :
Dans cette définition, on confond la cause '*événement*' et son effet '*Traitement désiré* ou **RINT**' : abusivement, cause & routine sont tous deux dits interruption.

(10)- INTRODUCTION à l' ASSEMBLEUR 'x86' ***(V)***

LES INTERRUPTIONS

Syntaxe générale (*int logicielles*) ::

INT *nombre*

'INT' : est le mot clé en ASM pour demander l'exécution d'une ISR
'nombre' : indique le numéro de cette ISR

(10)- INTRODUCTION à l' ASSEMBLEUR 'x86' ***(V)***

LES INTERRUPTIONS

Syntaxe générale (*int logicielles*) ::

INT *nombre*

1

- **INT** : Keyword
- *Nombre* [00-FFh] --(récupère)--> (CS : IP)_ISR--(accès)--> ISR

(***ISR*** : Interrupt Service Routine = ***RINT***)

'nombre' : (numéro de l' ISR) est compris entre 00 et FF en hexadécimal; il permettra de récupérer les valeurs d'adresses (CS & IP) propres à chaque ISR selon un schéma renseigné en slide 11.

(10)- INTRODUCTION à l' ASSEMBLEUR 'x86' ***(V)***

LES INTERRUPTIONS

Syntaxe générale (*int logicielles*) ::

INT *nombre*

1. • **INT** : Keyword
   • *Nombre* [00-FFh] --(récupère)--> (CS : IP)_ISR --(accès)--> ISR
   (***ISR*** : Interrupt Service Routine = ***RINT***)

2. (CS : IP)_ISR : Vecteur d'interruption (VI) :
   - situé à l'@ « 4*nombre »
   - pointe sur l'@ (CS & IP) de RINT= (CS : IP)_ISR

La valeur de (CS & IP) est dite VI (Vecteur d'Interruption): elle permet, après multiplication par 4 (car CS & IP nécessitent 4 octets (pr le 80286), de récupérer l'adresse de position de l'ISR(voir *slide 11*).

(10)- INTRODUCTION à l' ASSEMBLEUR 'x86' ***(V)***

LES INTERRUPTIONS

Syntaxe générale (*int logicielles*) ::

INT *nombre*

1. • **INT** : Keyword
   • *Nombre* [00-FFh] --(récupère)--> (CS : IP)_ISR--(accès)--> ISR
   (***ISR*** : Interrupt Service Routine = ***RINT***)

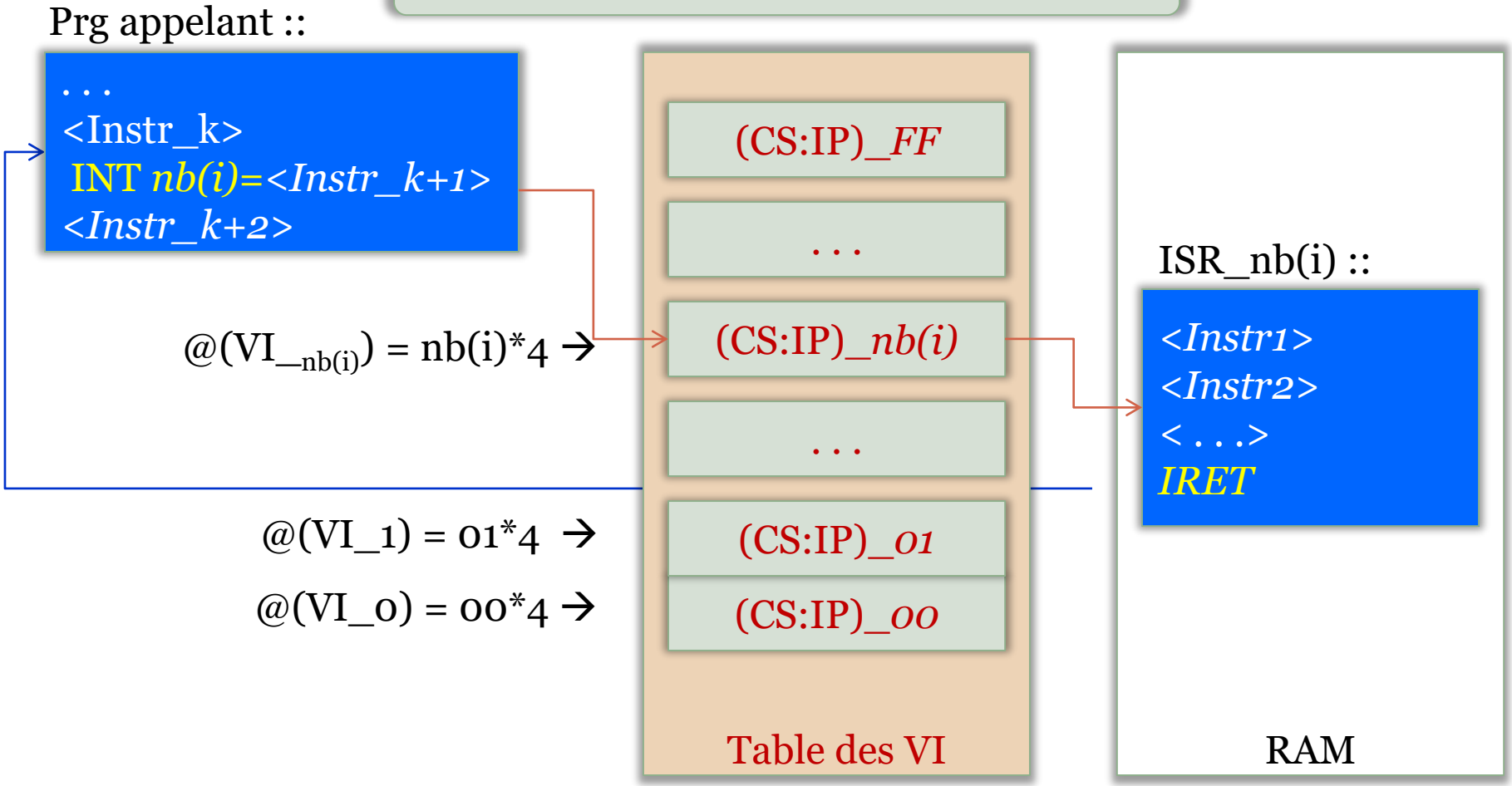
2. (CS : IP)_ISR : Vecteur d'interruption (VI) :
   - situé à l'@ « 4*nombre »
   - pointe sur l'@ (CS & IP) de RINT= (CS : IP)_ISR

3. Accessoires (paramétrage)::
   (AH) :: paramètres de l'ISR = ***Fonction*** de l'ISR
   (AL) :: paramètres de l'ISR= paramètre secondaire de l'ISR

(10)- INTRODUCTION à l' ASSEMBLEUR 'x86' ***(V)***

LES INTERRUPTIONS

Mécanisme d'accès



(10)- INTRODUCTION à l' ASSEMBLEUR 'x86' ***(V)***

LES INTERRUPTIONS

Dans le slide précédent:
L'appel d'interruption '*INT nb(i)*' qui est également l'instruction à l'ordre (k+1) *<Instr_k+1>* recquiert l'exécution de la routine d'interruption désignée par ISR_nb(i).
L'exécution de cette dernière nécessite le chargement de son adresse (CS:IP)_ISR dans les registres CS et IP du CPU.
La récupération de cette adresse (CS:IP)_ISR se fait en multipliant le numéro de l'interruption (ici : *nb(i)*) par 4, ce qui permet de charger CS_ISR et IP_ISR depuis la table des VI (Vecteurs d'Interruptions) située en zone spécifique de la RAM.
Les étapes successives de ce mécanisme sont explicitées dans le *slide* suivant:

(10)- INTRODUCTION à l' ASSEMBLEUR 'x86' ***(V)***

LES INTERRUPTIONS

Mécanisme d'accès :: **résumé**

Exécution de:

INT *nb(i)* =>

1. *CS:IP (Instr_k+2)* → *PILE*
2. *Accès à* Table des VI :: [@($VI_{nb(i)}$) = nb(i)*4] & récupération du contenu (CS:IP)_*nb(i)*
3. *Chargement* (CS:IP)_*nb(i)* → [CS : IP]_CPU (donc exécution de *ISR_nb(i)*)
4. Retour (*IRET*) à l'instruction < *Instr_k+2* > *du* «Prg appelant » après dépilement (voir 1)

(10)- INTRODUCTION à l' ASSEMBLEUR 'x86' ***(V)***

LES INTERRUPTIONS

Dans le *slide* qui suit:
Certains exemples d'interruptions très fréquents est présenté dans un tableau de synthèse. On y retrouve:
Dans la colonne **'CODE'** la syntaxe et le numéro de l'INT.
Dans la colonne **'CLASSIFICATION'** le domaine d'appartenance de l'INT (DOS ou BIOS).
Dans la colonne **'FONCTION'** certaines fonctionnalités de l'INT; le choix parmi les différentes fonctionnalités sa fait en utilisant la paramétrage approprié (dans AH, et éventuellement AL et DX).

(10)- INTRODUCTION à l' ASSEMBLEUR 'x86' ***(V)***

LES INTERRUPTIONS

EXEMPLES :: Interruptions fréquentes

CODE	Classification	Fonction
INT 10h	**BIOS**	*Affichage 'vidéo'*:: position curseur, balayage, graphiques . . .
INT 16h		Acquisition '*clavier*' :: lecture clavier, état clavier, gestion buffer . . .
INT 17h	**DOS**	'*Imprimante*': : Init, Etat, gestion buffer, . .
INT 1Ah		'*Horloge*' :: comptage et lecture « *date & heure* »
INT 21h		'*Fcts DOS*' :: 90 fcts « E/S, gestion fichier, gestion mémoire, . . »

(10)- INTRODUCTION à l' ASSEMBLEUR 'x86' ***(V)***

LES INTERRUPTIONS

ISR :: Exemple :: ***'Get Date Fct'***

APPEL :: INT 21H , fonction '2AH'

```
MOV AH, 2AH
INT 21H
MOV Year, CX
MOV Month, DH
MOV Day, DL
MOV DayOfWeek, AL
```

L'appel « **INT 21H** » avec le paramétrage '**AH = 2Ah** ' permet de charger (par le DOS) les valeurs systèmes respectives de l'année, du mois, et du jour dans CX, DH puis DL; ces valeurs sont ici récupérées et copiées dans les variables 'Year ', 'Month' et 'Day'.

(10)- INTRODUCTION à l' ASSEMBLEUR 'x86' ***(V)***

LES INTERRUPTIONS

ISR :: Exemple :: ***'Set Date Fct'***

APPEL :: INT 21H , fonction '2BH'

```
MOV AH, 2BH
MOV CX, Year
MOV DH, Month
MOV DL, Day
INT 21H
CMP AL, 0
JNE badDate
```

L'appel « **INT 21H** » avec le paramétrage '**AH = 2Bh** ' effectue la fonction inverse de la fonction '**AH = 2Ah** ' puisqu'elle permet, après chargement des valeurs année, mois puis jour dans les variables 'Year ', 'Month' et 'Day', de les charger dans les ressources systèmes: ce dernier, lors de l'appel « int 21 », charge en mémoire, les valeur contenues dans CX, DH puis DL respectivement.

(10)- INTRODUCTION à l' ASSEMBLEUR 'x86' ***(V)***

LES INTERRUPTIONS

F I N