جامعة مولاي إسماعيل
UNIVERSITÉ MOULAY ISMAÏL

Pr: Ismaili Alaoui El Mehdi A.U: 2023/2024

Architecture des ordinateurs

Chapitre 6: Interruptions & Chaînes de caractères

1 Interruptions

1. 1 Introduction

— Une interruption termine l'exécution séquentielle des instructions par le microprocesseur.
— Lors d'une interruption, l'exécution du programme principal est suspendue.
— Lorsque l'interruption est exécutée le programme principal est continué.

❒ Exemple:
Lorsque la souris est déplacée, le programme en cours est suspendue pendant un bref instant pour gérer ce déplacement.

Q. Comment fait-on pour écrire une chaîne de caractères à l'écran? Ou bien pour lire un caractére entré au clavier?

➥ On peut faire appel à des fonctions (affichage, saisie, ...) du MS-DOS à partir d'un programme écrit en langage assembleur, grâce à l'instruction INT (interruption) suivie du numéro de l'interruption;
➥ Sur le PC, les interruptions sont spécifiées par un numéro sur 8 bits : il y'a donc 256 interruptions différentes (int 00h à int FFh);
➥ L'interruption 21h (INT 21h) offre un regroupement de tous les services offert par le MS-DOS;
➥ Le numéro de la fonction requise doit être placé dans le registre AH (AH allant de 00H à 71H);

1. 2 La fonction 02H

➥ Rôle: Affichage d'un caractère.
➥ Entrées:
— AH = 02h
— DL = le code ASCII du caractère à afficher

➥ Sortie: Affichage du caractère.

❒ Exemple:
MOV DL, 41h=65
MOV AH, 02h ; fonction no. 2
INT 21H ; appel au MS-DOS
⇒ Affichage de "A" sur l'écran

1. 3 La fonction 09H

➥ Rôle: Affichage d'une chaîne de caractères.
➥ Entrées:
— AH = 09h
— DX = offset de la chaîne de caractères
➥ Sortie: Affichage de la chaîne de caractères.

❒ Exemple:
...
msg: db "bonjour$"; la chaîne doit se terminer par un "$ ou 24h",
;qui ne sera pas affiché
...
MOV DX, msg; DX pointe vers l'adresse du premier caractère de la
;chaîne de caractères msg
MOV AH, 09h ; fonction no. 9
INT 21h ; appel au MS-DOS
⇒ Affichage de "bonjour" sur l'écran.

1. 4 La fonction 08H

➥ Rôle: Saisie d'un caractère au clavier sans sortie (sans écho).
➥ Entrée:
— AH = 08h
➥ Sortie:
— AL = Le code ASCII du caractère lu

❒ Exemple:
MOV AH, 08h ; fonction no. 01h
INT 21h
AL ← Le code ASCII du caractère lu.

❒ Remarque:
La fonction 01h : Même principe que la fonction 08h, mais avec une sortie (avec écho).

1. 5 La fonction 0AH

➥ Rôle:Saisie d'une chaîne de caractères.
➥ Entrées:
— AH = 0Ah
— DX = Adresse d'offset du buffer (mémoire tampon)
➥ Sortie: La chaîne de caractère lue est placée dans un buffer à l'adresse DS:DX.

❒ Exemple:

```
txt: db 32 dup('$')
mov dx,txt
mov ah,0Ah
int 21h
```

❒ Remarques:

1. Le nombre de caractères réellement lus se trouve dans le 2ème octet de DX.
2. La chaine de caractères se trouve dans DX à partir du 3ème octet.

❒ Exercice d'application:
Écrire une séquence d'instructions qui

1. lit une chaîne de caractères, la mémorise dans une variable txt;
2. range dans cx la taille de chaîne de caractère lue;
3. range dans al le 1er caractère la chaîne de caractère lue.

.....
.....
.....
.....
.....

.....
.....
.....

1. 6 La fonction 4CH

➥ Rôle: Terminer un programme.
➥ Entrée:
— AH = 4Ch
➥ Sortie:
— Fin du programme

❒ Exemple:

```
...
MOV AH, 4CH
INT 21H
```

➥ À mettre à la fin de chaque fin programme; c'est l'équivalent du return (0) en C. Ces instructions ont pour effet de retourner au MS-DOS

2 Chaînes de caractères

2. 1 Introduction

— Une chaîne de caractère est un tableau de type octets ou mots. Les instructions de traitement de chaînes utilisent uniquement l'adressage indexé.
➥ Les registres utilisés sont : pour la source DS:SI, et pour la destination ES:DI.
— A chaque exécution d'une instruction de traitement de chaîne, les registres d'indexes sont incrémentés ou décrémentés selon le flag DF de registre d'état.

1. Si DF est placé à 1

➥ SI et DI sont décrémentés et les chaînes sont traitées de droite à gauche.

2. Si DF est placé à 0

➥ SI et DI sont incrémentés et les chaînes sont traitées de gauche à droite.

❒ Remarque:
— CLD: Positionner l'indicateur DF (Direction Flag) à 0.
Syntaxe : CLD ; sans opérande DF $\leftarrow$ 0.

— STD: Positionner l'indicateur DF à 1.
Syntaxe : STD ; sans opérande DF ← 1.

2. 2 Load String (LODSB, LODSW)

Chargement d'un Byte ou un word depuis DS:SI dans AL ou AX, plus la mise à jour de l'indexe SI selon DF. Le registre d'état reste inchangé.
Syntaxe : LODS(B ou W) ; sans opérande.

Le cas de AL :	Le cas de AX :
LODSB; AL ← [DS:SI]. ; SI ← SI± 1 selon DF.	LODSW; AX ← [DS :SI]. ; SI ← SI±2 selon DF.

❒ Exemple: Multiplier par 2 tous les éléments d'un tableau d'octet.

2. 3 Store String (STOSB, STOSW)

Copie le contenu de AL ou AX dans la case mémoire pointer par ES:DI plus la mise à jour de l'indexe DI selon DF. Le registre d'état reste inchangé.
Syntaxe : STOS(B ou W); sans opérande.

Le cas de AL :	Le cas de AX :
STOSB; [ES:DI] ← AL. ; DI ← DI± 1 selon DF.	STOSW; [ES:DI] ← AX. ; DI ← DI±2 selon DF.

❒ Exemple: Diviser par 2 tous les éléments d'un tableau des mots.

2. 4 Scan String (SCASB, SCASW)

Comparaison entre la case mémoire pointer par ES:DI avec AL ou AX, plus la mise à jour de l'indexe DI selon DF et le registre d'état.
Syntaxe : SCAS(B ou W); sans opérande.

Le cas de AL :	Le cas de AX :
SCASB ; CMP [ES:DI], AL ; DI ← DI± 1 selon DF. ; mise à jour de registre d'état.	SCASW ; CMP [ES:DI], AX ; DI ←DI± 2 selon DF. ; mise à jour de registre d'état.

❒ Exemple: Copier dans BL le nombre d'occurrence de 'e' dans une chaînes de caractères.

2. 5 Move String (MOVSB, MOVSW)

Copie le contenu de la case mémoire pointer par DS:SI dans la case mémoire pointer par ES :DI, plus la mise à jour les indexes SI et DI selon DF. Le registre d'état reste inchangé.

Syntaxe : MOVS(B ou W); sans opérande.

Le cas d'une case mémoire 8 bits:	Le cas d'une case mémoire 16 bits :
MOVSB; [ES:DI] ← [DS:SI]	MOVSW; [ES:DI] ← [DS:SI]
; SI ← SI± 1 selon DF.	; SI ← SI± 2 selon DF.
; DI ← DI± 1 selon DF.	; DI ← DI± 2 selon DF.

❒ Exemple: Copier un tableau de caractères dans un autre tableau.

2. 6 Compare String (CMPSB, CMPSW)

Comparaison entre la case mémoire pointer par ES:DI avec la case mémoire pointer par DS : SI, plus la mise à jour des indexes SI, DI selon DF et le registre d'état.
Syntaxe : CMPS(B ou W); sans opérande.

Le cas d'une case mémoire 8 bits:	Le cas d'une case mémoire 16 bits:
CMPSB; CMP [ES:DI],[DS:SI]	MOVSW; CMP [ES:DI],[DS:SI]
; SI ← SI± 1 selon DF.	; SI ← SI± 2 selon DF.
; DI ← DI± 1 selon DF.	; DI ← DI± 2 selon DF.
; mise à jour de registre d'état.	; mise à jour de registre d'état.

❒ Exemple: Comparer deux chaînes ont la même taille.

2. 7 Préfixes

2. 7. 1 Préfixe REP

REP décrément automatiquement CX est test est ce qu'il est égal à zéro ou non. Si CX = 0 REP s'arrêt. Les instructions qui utilisent REP sont : MOVS, LODS et STOS.

❒ Algorithme de fonctionnement:
Si CX $\neq$ 0 alors
. 1- On exécute l'instruction de traitement de chaîne.
. 2- CX $\leftarrow$ CX - 1.
. 3- Répéter.
Sinon
. Sort de la boucle de répétition.
Fin si.

❒ Exemple: Initialiser un tableau avec N caractères 'A'.

.....
.....
.....
.....
.....

2. 7. 2 Préfixe REPE et REPZ

Dérivé de REP, cette instruction continu la répétition tant que CX $\neq$ 0 et ZF=1. Cette instruction est très intéressante pour comparer si deux chaînes sont identiques.

❒ Algorithme de fonctionnement:
Si CX $\neq$ 0 alors
. 1- On exécute l'instruction de traitement de chaîne.
. 2- CX $\leftarrow$ CX-1.
. 3- Si ZF=1 alors répéter.
. Sinon
. Sort de la boucle de répétition.
. Fin si
Sinon
. Sort de la boucle de répétition.
Fin si.

❒ Exemple: Comparer deux chaînes ont la même taille.

.....
.....
.....
.....
.....
.....
.....
.....

2. 7. 3 Préfixe REPNE et REPNZ

Toujours un dérivé de REP, cette instruction continu la répétition tant que CX $\neq$ 0 et ZF=0.

❒ Algorithme de fonctionnement:
Si CX $\neq$ 0 alors
. 1- On exécute l'instruction de traitement de chaîne
. 2- CX $\leftarrow$ CX-1
. 3- Si ZF=0 alors répéter
. Sinon
. Sort de la boucle de répétition
. Fin si
Sinon
. Sort de la boucle de répétition
Fin si

❒ Exemple: Recherche de la lettre 'B' dans une chaîne de caractères.

.....
.....
.....
.....
.....
.....
.....
.....