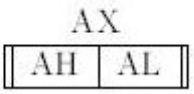
Assembleur : Systèmes x86

I. Registres 16 bits du 8086 :

Le processeur 8086 original fournissait quatre registres généraux de 16 bits décomposables en 2 registres de 8 bits pouvant être utilisés comme des registres d'un octet indépendants.

Le registre AX pouvait être décomposé en AH et AL. Le registre AH contient les 8 bits de poids fort de AX et AL contient les 8 bits de poids faible. Souvent, AH et AL sont utilisés comme des registres d'un octet indépendants ; cependant, il est important de réaliser qu'ils ne sont pas indépendants de AX. Changer la valeur de AX changera les valeurs de AL et BL et vice versa. Les registres généraux sont utilisés dans beaucoup de déplacements de données et instructions arithmétiques.



Il y a deux registres d'index de 16 bits : SI et DI. Ils sont souvent utilisés comme des pointeurs, mais peuvent être utilisés pour la plupart des mêmes choses que les registres généraux. Cependant, ils ne peuvent pas être décomposés en registres de 8 bits.

Les registres 16 bits BP et SP sont utilisés pour pointer sur des données dans la pile du langage machine et sont appelés le pointeur de base et le pointeur de pile, respectivement.

Les registres 16 bits CS, DS, SS et ES sont des registres de segment. Ils indiquent quelle zone de la mémoire est utilisée pour les différentes parties d'un programme. CS signifie Code Segment, DS Data Segment, SS Stack Segment (segment de pile) et ES Extra Segment. ES est utilisé en tant que registre de segment temporaire.

Le registre de pointeur d'instruction (IP) est utilisé avec le registre CS pour mémoriser l'adresse de la prochaine instruction à exécuter par le processeur.

Normalement, lorsqu'une instruction est exécutée, IP est incrémenté pour pointer vers la prochaine instruction en mémoire.

Le registre FLAGS stocke des informations importantes sur les résultats d'une instruction précédente. Ces résultats sont stockés comme des bits individuels dans le registre.

II. Registres 32 bits du 80386

Les processeurs 80386 et plus récents ont des registres étendus : EAX, EBX, ECX, EDX, ESI et EDI. Pour la compatibilité ascendante, AX fait toujours référence au registre 16 bits et on utilise EAX pour faire référence au registre 32 bits. AX représente les 16 bits de poids faible de EAX tout comme AL représente les 8 bits de poids faible de AX (et de EAX). Il n'y a pas moyen d'accéder aux 16 bits de poids fort de EAX directement.

La plupart des autres registres sont également étendus. BP devient EBP ; SP devient ESP ; FLAGS devient EFLAGS et IP devient EIP. Cependant, contrairement aux registres généraux et d'index, en mode protégé 32 bits seules les versions étendues de ces registres sont utilisées.

Les registres de segment sont toujours sur 16 bits dans le 80386. Il y a également deux registres de segment supplémentaires : FS et GS. Leurs noms n'ont pas de signification particulière. Ce sont des segments temporaires supplémentaires (comme ES).

- **Remarque**

Une des définitions du terme mot se réfère à la taille des registres de données du processeur. Pour la famille du 80x86, le terme est désormais un peu confus. Dans le Tableau 1.2, on voit que le terme mot est déni comme faisant 2 octets (ou 16 bits). Lorsque le premier 8086 est apparu. Lorsque le 80386 a été développé, le terme mot est déni comme faisant 2 octets (ou 16 bits). Il a été décidé de laisser la définition de mot inchangée, même si la taille des registres avait changé.

III. Retour sur le mode réel et protégé

En mode réel, la mémoire est limitée à seulement un mégaoctet (220 octets).

Les adresses valides vont de 00000 à FFFFF (en hexa). Ces adresses nécessitent un nombre sur 20 bits. Cependant, un nombre de 20 bits ne tiendrait dans aucun des registres 16 bits du 8086. Intel a résolu le problème, en utilisant deux valeurs de 16 bits pour déterminer une adresse. La première valeur de 16 bits est appelée le sélecteur. Les valeurs du sélecteur doivent être stockées dans des registres de segment. La seconde valeur de 16 bits est appelée le déplacement (offset). L'adresse physique identifiée par un couple sélecteur/déplacement 32 bits est calculée par la formule

16 * sélecteur + déplacement

De fait, la valeur du sélecteur est un numéro de paragraphe.

Les adresses réelles segmentées ont des inconvénients :

- une seule valeur de sélecteur peut seulement référencer 64Ko de mémoire (la limite supérieure d'un déplacement de 16 bits). Que se passe-t-il si un programme a plus de 64 Ko de code ? Une seule valeur de CS ne peut pas être utilisée pour toute l'exécution du programme. Le programme doit être divisé en sections (appelées segments) de moins de 64 Ko. Lorsque l'exécution passe d'un segment à l'autre, la valeur de CS doit être changée. Des problèmes similaires surviennent avec de grandes quantités de données et le registre DS. Cela peut être très gênant ! ;

- chaque octet en mémoire n'a pas une adresse segmentée unique. L'adresse physique 04808 peut être référencée par 047C:0048, 047D:0038, 047E:0028 ou 047B:0058. Cela complique la comparaison d'adresses segmentées.

IV. Mode protégé 16 bits :

Dans le mode protégé 16 bits du 80286, les valeurs du sélecteur sont interprétées de façon totalement différente par rapport au mode réel. En mode réel, la valeur d'un sélecteur est un numéro de paragraphe en mémoire.

En mode protégé, un sélecteur est un indice dans un tableau de descripteurs. Dans les deux modes, les programmes sont divisés en segments. En mode réel, ces segments sont à des positions fixes en mémoire et le sélecteur indique le numéro de paragraphe auquel commence le segment. En mode protégé, les segments ne sont pas à des positions fixes en mémoire physique. De fait, ils n'ont même pas besoin d'être en mémoire du tout !

Le mode protégé utilise une technique appelée mémoire virtuelle. L'idée de base d'un système de mémoire virtuelle est de ne garder en mémoire que les programmes et les données actuellement utilisés. Le reste des données et du code sont stockés temporairement sur le disque jusqu'à ce qu'on ait à nouveau besoin d'eux. Dans le mode protégé 16 bits, les segments sont déplacés entre la mémoire et le disque selon les besoins. Lorsqu'un

segment est rechargé en mémoire depuis le disque, il est très probable qu'il sera à un endroit en mémoire différent de celui où il était avant d'être placé sur le disque.

Tout ceci est effectué de façon transparente par le système d'exploitation. Le programme n'a pas à être écrit différemment pour que la mémoire virtuelle fonctionne.

En mode protégé, chaque segment est assigné à une entrée dans un tableau de descripteurs. Cette entrée contient toutes les informations dont le système a besoin à propos du segment. Ces informations indiquent : s'il est actuellement en mémoire ; s'il est en mémoire, où il se trouve ; les droits d'accès (p.e., lecture seule). L'indice de l'entrée du segment est la valeur du sélecteur stockée dans les registres de segment.

Un gros inconvénient du mode protégé 16 bits est que les déplacements sont toujours des quantités sur 16 bits. En conséquence, les tailles de segment sont toujours limitées au plus à 64 Ko. Cela rend l'utilisation de grands tableaux problématique.

V. Mode protégé 32 bits

Le 80386 a introduit le mode protégé 32 bits. Il y a deux différences majeures entre les modes protégés 32 bits du 386 et 16 bits du 286 :

1. Les déplacements sont étendus à 32 bits. Cela permet à un déplacement d'aller jusqu'à 4 milliards. Ainsi, les segments peuvent avoir des tailles jusqu'à 4 Go ;

2. Les segments peuvent être divisés en unités plus petites de 4 Ko appelées pages. Le système de mémoire virtuelle fonctionne maintenant avec des pages plutôt qu'avec des segments. Cela implique que seules certaines parties d'un segment peuvent être présentes en mémoire à un instant donné. En mode 16 bits du 286, soit le segment en entier est en mémoire, soit rien n'y est. Ce qui n'aurait pas été pratique avec les segments plus grands que permet le mode 32 bits.

Dans Windows 3.x, le mode standard fait référence au mode protégé 16 bits du 286 et le mode amélioré (enhanced) fait référence au mode 32 bits.

Windows 9X, Windows NT/2000/XP, OS/2 et Linux fonctionnent tous en mode protégé 32 bits paginé.

VI. Interruptions

Quelques fois, le flot ordinaire d'un programme doit être interrompu pour traiter des évènements qui requièrent une réponse rapide. Le matériel d'un ordinateur offre un mécanisme appelé interruptions pour gérer ces évènements.

Par exemple, lorsqu'une souris est déplacée, la souris interrompt le programme en cours pour gérer le déplacement de la souris (pour déplacer le curseur, etc.). Les interruptions provoquent le passage du contrôle à un gestionnaire d'interruptions. Les gestionnaires d'interruptions sont des routines qui traitent une interruption. Chaque type d'interruption est assignée à un nombre entier. Au début de la mémoire physique, réside un tableau de vecteurs d'interruptions qui contient les adresses segmentées des gestionnaires d'interruptions. Le numéro d'une interruption est essentiellement un indice dans ce tableau.

Les interruptions externes proviennent de l'extérieur du processeur (la souris est un exemple de ce type). Beaucoup de périphériques d'E/S soulèvent des interruptions (p.e., le clavier, le timer, les lecteurs de disque, le CD-ROM et les cartes son). Les interruptions internes sont soulevées depuis le processeur, à cause d'une erreur ou d'une instruction d'interruption.

Beaucoup de gestionnaires d'interruptions redonnent le contrôle au programme interrompu lorsqu'ils se terminent. Ils restaurent tous les registres aux valeurs qu'ils avaient avant l'interruption. Ainsi, le programme interrompu s'exécute comme si rien n'était arrivé (excepté qu'il perd quelques cycles processeur).