

Concours de saut sur TI 58/59 étiquettes ou adressage numérique ?

En programmation, les chemins les plus courts sont toujours les meilleurs.

Ce sont donc eux qu'il faut emprunter pour faire des sauts à l'intérieur d'un programme.

■ Les possesseurs de TI 58/59 connaissent bien les deux possibilités de branchement à une adresse donnée : on peut renvoyer à une étiquette (GTO INV par exemple) ou à une adresse numérique (GTO 150).

La difficulté, quand on a deux options comme celles-ci, ce n'est pas tellement d'en choisir une, mais plutôt de les mélanger dans la meilleure proportion et de le faire à bon escient. Il n'est donc pas question de chercher à savoir si l'un des deux adressages est meilleur que l'autre ; l'important est de connaître leurs avantages et leurs inconvénients respectifs pour en tirer parti dans la construction des programmes.

Voyons d'abord ce qui se passe lors d'un saut effectué à la suite d'un test ($x=t$; $x \geq t$), d'une boucle (Dsz), ou d'un branchement inconditionnel (GTO). Lorsque l'adresse est une étiquette, le pointeur la recherche à partir de la ligne 000. Cette recherche n'est pas instantanée : il faut compter environ une

seconde pour une étiquette placée à la ligne 450.

Si l'adresse est numérique au contraire, le branchement sera presque immédiat, d'où un gain de temps non négligeable.

L'adressage numérique a donc un net avantage en ce qui concerne la vitesse d'exécution, ce qui laisse penser, a priori, qu'un programme *dans sa version définitive* devra utiliser ce type d'adressage.

Je viens d'exposer l'argument le plus important en faveur de l'adressage numérique. L'étiquetage a, bien entendu, lui aussi un point fort qui le rend très précieux au moment de la construction d'un programme. Pour renvoyer à ce numéro de ligne, en effet, il faut le connaître, et ce n'est pas toujours possible quand cette ligne se situe en aval du branchement. Si nous décidons, à la ligne 100, d'un branchement vers un bloc qui commence aux alentours de la ligne 150, nous devons attendre d'avoir écrit cette ligne pour en connaître exactement le numéro.

L'étiquetage nous affranchit de cette gymnastique en permettant un branchement à une étiquette sans avoir à nous soucier de sa position dans le programme. De plus, les modifications nécessitant l'insertion ou la destruction de lignes changent la numérotation de toutes celles qui sont en aval de la modification. On entrevoit aisément le travail à accomplir pour changer en conséquence toutes les adresses de saut si l'on utilise l'adressage numérique.

Il est donc souhaitable, sinon

indispensable, d'employer les étiquettes lors de la construction et la mise au point d'un programme.

Une fois le programme terminé, et dûment étiqueté, il semblerait logique de remplacer les étiquettes par des adresses numériques. Dans la pratique, je vous propose un compromis : plutôt que de conserver un seul programme, unique et définitif, gardez d'abord une version du programme étiqueté (liste ou carte magnétique), et une version utilisant à votre gré et de façon mixte les deux types d'adressage.

- La version étiquetée se révèle utile dans plusieurs cas : le programme peut avoir à subir des modifications, améliorations ou ajouts ultérieurs. On comprend que ces transformations réclameront beaucoup de travail sur un programme "rigidifié" par l'adressage numérique. La compréhension du programme par un tiers est grandement facilitée par la présence de blocs bien apparents grâce à la dénotation "LbI".

- La version définitive, c'est-à-dire celle qui sera utilisée en pratique, ne sera pas nécessairement "numérique" du début jusqu'à la fin. Un programme dont l'exécution demande trois secondes en version étiquetée ne gagnera pas beaucoup à être transformé.

Plus précisément, on connaît les "dévoreurs de temps" dans un programme : ce sont les boucles ; on aura donc souvent intérêt à les "numériser".

Un dernier point impose parfois l'un ou l'autre des deux adressages :

Premier temps de la manœuvre : on note + 1 tous les branchements vers des étiquettes et - 2 toutes les étiquettes.

	000	76	LBL	
	001	11	A	
	002	05	5	
	003	66	PAU	
	004	42	STD	
	005	01	01	
	006	61	GTO	+1
	007	22	INV	→
	008	76	LBL	
-2	→	009	24	CE
	010	01	1	
	011	00	0	
	012	42	STD	
	013	01	01	
	014	76	LBL	
-2	→	015	22	INV
	016	43	RCL	
	017	01	01	
	018	66	PAU	
	019	97	DSZ	
	020	01	01	
	021	22	INV	+1
	022	61	GTO	
	023	24	CE	+1
	001	11	A	
	010	009	24	CE
	016	015	22	INV

c'est la limitation en lignes de programme. Phénomène peu connu sur TI 59, il se rencontre souvent sur TI 58, et il s'en faut parfois d'un rien pour qu'un programme rentre ou ne rentre pas dans la mémoire.

Faisons nos comptes :

- la déclaration d'étiquette " coûte " deux lignes (Lbl INV),
- le branchement à une étiquette coûte deux lignes (GTO INV),
- le branchement à une adresse numérique coûte trois lignes (GTO 01 50 ; pour GTO 150).

On gagne donc à ce qu'une adresse vers laquelle il n'existe qu'un seul branchement soit numérique (pas de déclaration, saut : trois lignes, coût total : trois lignes).

En revanche, si l'on a un branchement fréquent vers une même adresse, on aura intérêt à utiliser une étiquette. En effet, si la déclaration coûte deux lignes, chaque branchement économise une ligne par rapport à l'adressage numérique. Il

suffit alors de placer les étiquettes concernées en début de programme pour limiter le temps de recherche.

Pour conclure, voici une méthode rapide et rationnelle pour " numériser " un programme étiqueté. Le programme utilisé comme exemple ne trouve pas une application évidente (je me demande moi-même à quoi il peut servir), mais le but en est essentiellement pédagogique.

Procédé :

- écrire la liste du programme avec les numéros de lignes (c'est ici, entre autres, que l'imprimante peut rendre des services) ;
- repérer quelles sont les déclarations d'étiquettes qui devront être " numérisées " et placer une flèche à gauche devant la ligne qui suit la déclaration (début du bloc proprement dit) ; les utilisateurs de PC 100 exécuteront 2nd Op 08 qui donne la liste des étiquettes et ils ajouteront 1 à chaque numéro de ligne ;
- placer une flèche à droite au

	000	76	LBL		
	001	11	A		
	002	05	5		
	003	66	PAU		
	004	42	STD		
	005	01	01		
	006	61	GTO	+1	
	007	22	INV	→	
	008	76	LBL		
	009	24	CE		
-2	→	010	01	1	
	011	00	0	Total: -1	
	012	42	STD		
	013	01	01		
	014	76	LBL		
-2	→	015	22	INV	
	016	43	RCL	Total: -3	
	017	01	01		
	018	66	PAU		
	019	97	DSZ		
	020	01	01	+1	
	021	22	INV	→	
	022	61	GTO	+1	
	023	24	CE	→	
	001	11	A		
	009	010	009	24	CE
	013	016	015	22	INV

Deuxième étape : on fait le décompte des pas en plus et en moins.

niveau de chaque branchement devant être " numérisé " ;

- au-dessus de chaque flèche correspondant à une déclaration, écrire " -2 " ;
- au-dessus de chaque flèche correspondant à un branchement, écrire " +1 " (voir liste 1).

Dès lors, le travail est simple. En partant du début du programme, faire la somme des +1 et des -2. Arrivé à une déclaration d'étiquette, faire le total et l'ajouter au numéro correspondant à cette étiquette. Continuer ainsi jusqu'à la fin, et l'on a alors toutes les adresses numériques des étiquettes devant disparaître (voir liste 2).

	000	76	LBL	
	001	11	A	
	002	05	5	
	003	66	PAU	
	004	42	STD	
	005	01	01	
	006	61	GTO	
	007	00	00	
	008	13	13	
	009	01	1	
	010	00	0	
	011	42	STD	
	012	01	01	
	013	43	RCL	
	014	01	01	
	015	66	PAU	
	016	97	DSZ	
	017	01	01	
	018	00	00	
	019	13	13	
	020	61	GTO	
	021	00	00	
	022	09	09	

Le programme définitif " figé " avec ses adressages numériques.

La fin du travail consiste à détruire les déclarations d'étiquettes (Delete), et à remplacer chaque saut vers une étiquette par un branchement à l'adresse numérique correspondante, sans oublier d'insérer une ligne à chaque branchement (voir liste 3).

Voilà, les explications paraissent peut-être un peu longues, mais à l'usage cette méthode se révèle facile à comprendre et très pratique.

□ Antoine Jennet