

**EDILE**

**manuel de référence**

02 SEP 1990

**SEMS**

FONDS DOCUMENTAIRE

SMP

ÉDITEUR DE LIENS

ÉDILE

MANUEL DE RÉFÉRENCE

SOMMAIRE		Pages
1 – INTRODUCTION		1
1.1 – PRÉSENTATION DE L'ÉDITEUR DE LIENS		1
1.2 – CARACTÉRISTIQUES DE L'ÉDITEUR DE LIENS		1
1.3 – FONCTIONNEMENT DE L'ÉDITEUR DE LIENS		2
1.3.1 – TRANSLATION DES ADRESSES		2
1.3.2 – RÉOLUTION DES RÉFÉRENCES EXTERNES		2
1.4 – STRUCTURE DES PROGRAMMES SOURCE		3
1.4.1 – FORMATS BINAIRES : TRANSLATABLE ET "LINK EDITABLE"		3
1.4.2 – COMPILATEUR PL 1600		4
2 – UTILISATION DE L'ÉDITEUR DE LIENS		6
2.1 – ENTRÉES-SORTIES		6
2.2 – ACTIVATION DE L'ÉDITEUR DE LIENS		6
2.2.1 – COMMANDE ILNK		6
2.2.2 – COMMANDE CLNK		7
2.2.3 – COMMANDE RLNK		7
2.2.4 – COMMANDE ELNK		8
2.2.5 – COMMANDE TLNK		8
2.2.6 – COMMANDE DLNK		9
2.2.7 – EXEMPLE D'UNE ÉDITION DE LIENS		9
2.3 – MESSAGES D'ERREUR		10
2.4 – INTERRUPTION D'UNE ÉDITION DE LIENS		11
2.5 – CHARGEMENT DE L'ÉDITEUR DE LIENS		11
2.5.1 – UTILISATION DU CHARGEUR AUTONOME		12
2.5.2 – UTILISATION DU CHARGEUR INTÉGRÉ A 80S		13
ANNEXE : LISTE DES NUMÉROS D'ERREUR		14

## 1 – INTRODUCTION

### 1.1 – PRÉSENTATION DE L'ÉDITEUR DE LIENS

L'éditeur de liens EDILE est un programme faisant partie du software de base du SOLAR 16. Sa fonction essentielle est la génération de programmes binaires acceptés par le chargeur translateur, à partir de programmes «link éditables» issus d'assemblages ou de compilations séparées.

Il permet à l'utilisateur de découper un programme en plusieurs unités, ce qui présente un certain nombre d'avantages:

- chaque unité peut être codée dans un langage de programmation adapté, et être assemblée ou compilée séparément.

- les communications entre unités sont rendues possibles par l'emploi de directives spécialisées, telles les directives de l'assembleur EXT et ENT. Un même symbole peut ainsi être défini dans une unité et référencé dans les autres unités composant le programme qui le déclarent alors «externe».

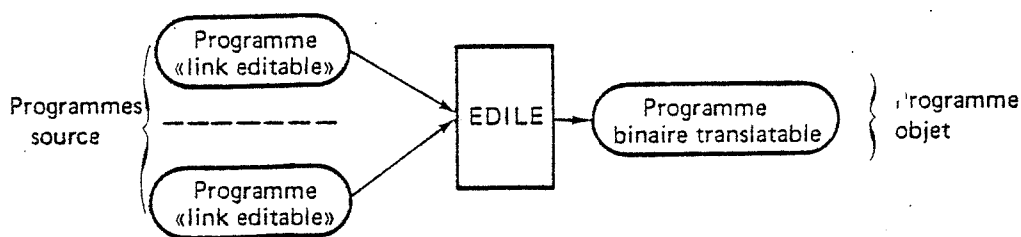
- seule l'unité en cours de mise au point nécessite d'être réassemblée ou recompilée lorsqu'une erreur est détectée. Le programme binaire ainsi obtenu est relié par l'éditeur de liens EDILE aux programmes binaires correspondant aux autres unités.

Découper un programme peut être rendu nécessaire lorsque la taille de la mémoire ne permet pas de l'assembler ou de le compiler en une seule fois (nombre de symboles trop important).

Les diverses unités peuvent utiliser les mêmes symboles «internes». Seuls les symboles «externes» nécessitent d'être uniques dans le programme.

### 1.2 – CARACTÉRISTIQUES DE L'ÉDITEUR DE LIENS

L'éditeur de liens fonctionne en un seul passage : EDILE délivre un programme objet en binaire translatable en ne faisant qu'une seule lecture de chaque programme source «link éditable».



Au cours de ce passage les opérations suivantes sont effectuées:

- vérification de la parité au niveau du caractère lu et du checksum au niveau du programme source, ce qui offre un maximum de sécurité
- redéfinition de toutes les adresses contenues dans les programmes source (constantes adresses, adresses d'implantation, etc...) en fonction de l'origine du programme objet
- réalisation des liaisons entre unités ayant été demandées, par exemple, par le jeu des directives d'assemblage EXT et ENT
- transmission des définitions de symboles externes, ce qui permet au programme objet de servir de source dans une édition ultérieure
- édition de la liste des symboles externes
- édition optionnelle des références non résolues
- édition optionnelle de la liste des symboles internes aux programmes source .

L'éditeur de liens EDILE s'exécute sous le contrôle du superviseur B0S. Il est prévu pour fonctionner avec la configuration minimale qui comprend:

- le calculateur équipé d'une mémoire 4K;
- un téléimprimeur.

### 1.3 – FONCTIONNEMENT DE L'ÉDITEUR DE LIENS

Les deux principales opérations réalisées par l'éditeur de liens sont la translation des adresses et la résolution des références externes.

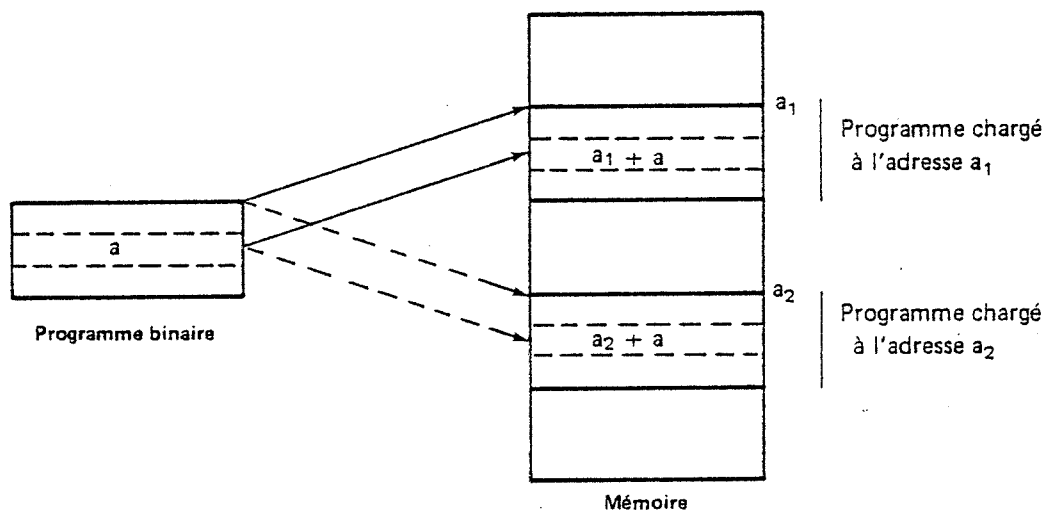
#### 1.3.1 – TRANSLATION DES ADRESSES

A partir des programmes symboliques qui leur sont fournis, l'assembleur ASM ou un compilateur produisent des programmes binaires dans lesquels les adresses sont « translatables ».

Ceci permet de ne déterminer l'implantation des programmes qu'au moment de leur chargement en mémoire.

Ils sont assemblés ou compilés à partir de l'adresse zéro et toutes les adresses sont calculées à partir de cette origine.

Supposons, par exemple, la valeur  $a$  affectée par l'assembleur à une constante adresse. Si le programme est chargé en mode maître à l'adresse  $a_1$  la constante subira une translation et sa valeur définitive sera  $a_1 + a$ .



Au cours d'une édition de liens EDILE réalisée sur les adresses les translations suivantes :

- les adresses contenues dans le premier programme rencontré sont inchangées
- à partir du second programme, les adresses sont incrémentées de la longueur totale des programmes traités auparavant.

Exemple :

- Programme  $P_1$  de longueur  $L_1$  : adresses inchangées
- Programme  $P_2$  de longueur  $L_2$  : adresses translatées de  $L_1$
- Programme  $P_3$  : adresses translatées de  $L_1 + L_2$

#### 1.3.2 – RÉOLUTION DES RÉFÉRENCES EXTERNES

Les communications entre programmes sont réalisées par le jeu de symboles externes qui sont des symboles globaux, les symboles internes à chaque programme étant alors locaux.

Au cours du traitement l'éditeur de liens gère une table où sont enregistrées toutes les définitions de symboles externes.

L'ordre dans lequel les définitions de symboles externes et les références à ces symboles apparaissent est indifférent, la définition d'un symbole pouvant précéder ou suivre toute référence à celui-ci.

## 1.4 – STRUCTURE DES PROGRAMMES SOURCES

### 1.4.1 – FORMATS BINAIRES : TRANSLATABLE ET «LINK EDITABLE»

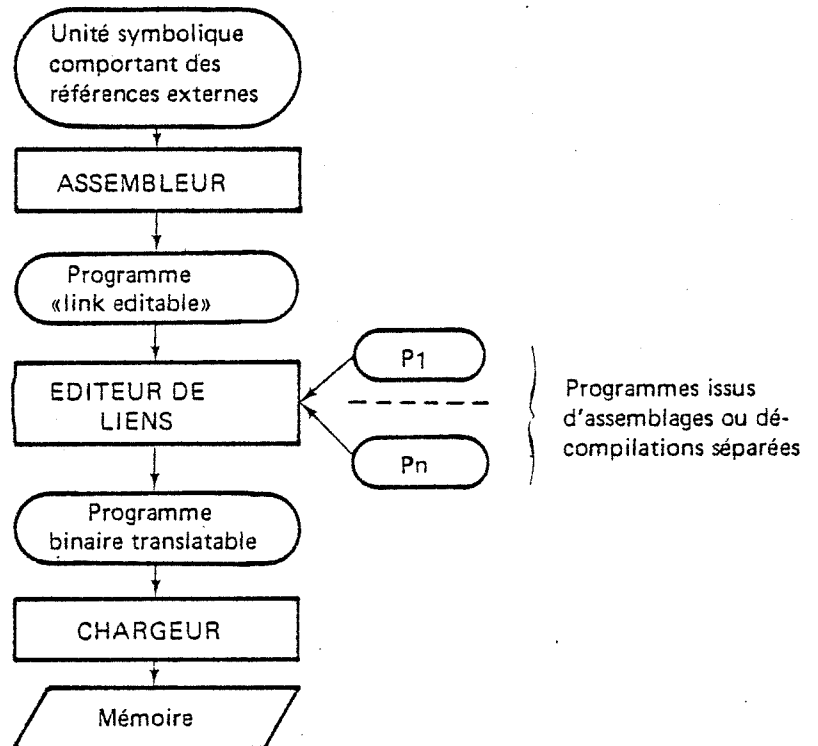
A partir d'un texte symbolique source l'assembleur peut générer un programme objet:

- en format binaire translatable, lorsque le texte source ne comporte aucune directive d'assemblage EXT
- en format binaire «link editable» dans le cas contraire.

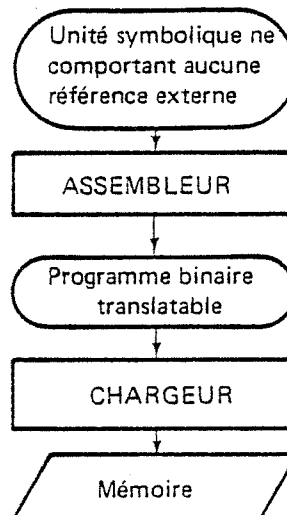
Le format binaire translatable est le seul accepté par le chargeur translateur.

La tentative de chargement d'un programme dans le format «link editable» se solde par un refus du chargeur qui imprime un message d'erreur.

a)



b)



**Remarque:**

Un programme binaire translatable issu de l'assembleur peut être «link édité» avec d'autres programmes issus d'assemblages ou de compilations séparées. Il est en effet «link editable». Il peut s'agir d'une unité indépendante ne réalisant aucune liaison avec les autres unités constituant le programme.

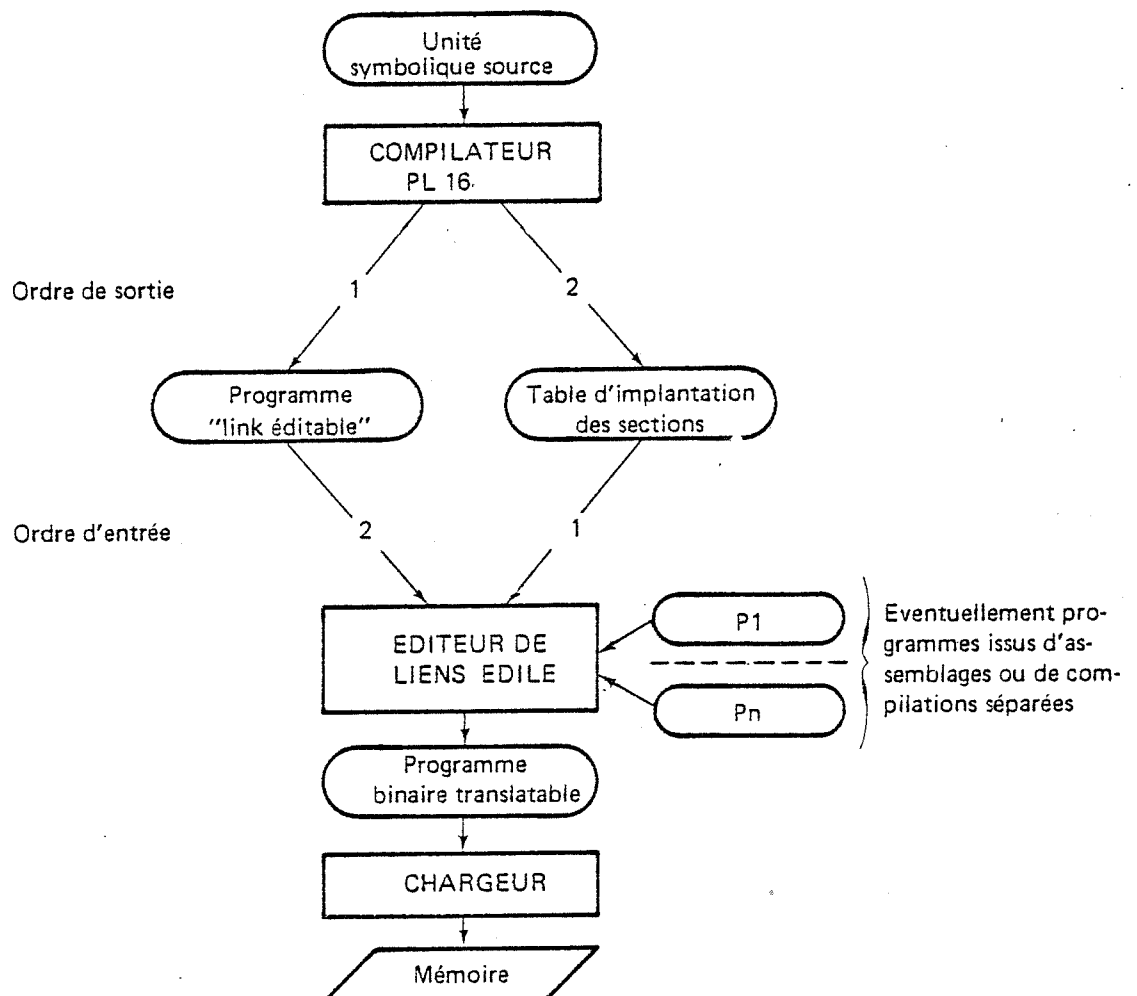
**1.4.2 - COMPILATEUR PL 16**

Pour chaque unité de compilation le compilateur PL 16 fournit dans l'ordre un programme «link editable», puis la table d'implantation des diverses sections composant l'unité.

Ces informations sont à fournir à l'éditeur de liens dans l'ordre inverse.

Il est à noter que, dans le cas d'un programme PL 16, le passage par l'éditeur de liens constitue une étape indispensable pour l'obtention d'un binaire translatable, seul format accepté par le chargeur.

Ce passage permet le calcul de toutes les adresses contenues dans le programme «link editable», à partir de la table d'implantation des sections fournie à l'éditeur de liens précédemment. En parallèle s'effectue l'impression de la liste des symboles internes définis dans l'unité.

**Remarque:**

L'éditeur de liens peut recevoir en entrée un certain nombre de programmes «link editables».

Ces programmes peuvent être présentés à EDILE dans un ordre quelconque.

La seule précaution à prendre est de faire précéder chaque programme «link editable» issu du compilateur PL 16 de la table d'implantation des sections qui lui est associée.

Si par exemple, les unités de compilation 1 et 2 ont fourni:

- programme «link editable» P1
- table d'implantation T1
- programme «link editable» P2
- table d'implantation T2 .

l'ordre de présentation à EDILE peut être indifféremment:

- T1, P1, T2, P2
- T2, P2, T1, P1

L'un quelconque des deux ordres :

- T1, P2, T2, P1
- T2, P1, T1, P2

serait détecté par l'éditeur de liens et donnerait lieu à un message d'erreur.



## 2 – UTILISATION DE L'ÉDITEUR DE LIENS

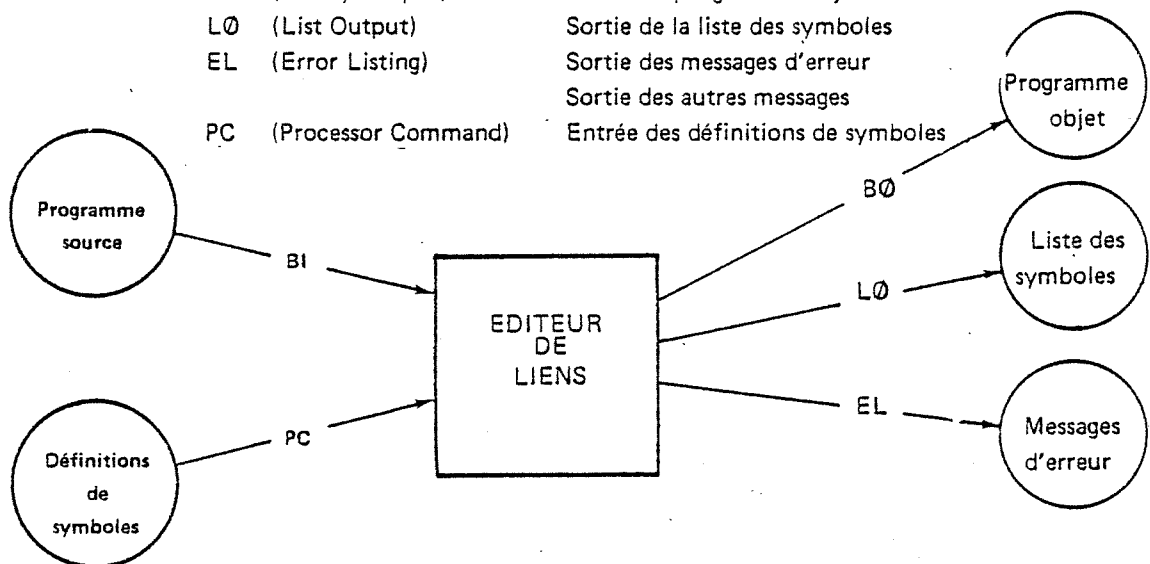
### 2.1 – ENTRÉES-SORTIES

L'éditeur de liens EDILE est un programme géré par le superviseur BØS.

Il utilise un certain nombre d'unités symboliques pour ses demandes d'entrées-sorties.

Ce sont les suivantes :

BI	(Binary Input)	Entrée des programmes sources.
BØ	(Binary Output)	Sortie du programme objet
LØ	(List Output)	Sortie de la liste des symboles
EL	(Error Listing)	Sortie des messages d'erreur
PC	(Processor Command)	Sortie des autres messages
		Entrée des définitions de symboles



Avant d'activer EDILE, l'utilisateur peut associer à chaque unité symbolique une unité fonctionnelle dont il dispose dans sa configuration.

Les commandes d'association sont décrites dans le manuel de référence de BØS.

### 2.2 – ACTIVATION DE L'ÉDITEUR DE LIENS

L'éditeur de liens EDILE possède 6 points d'entrée. A chacun d'eux correspond une commande qui est reconnue par le superviseur.

#### 2.2.1 – COMMANDE ILNK

La commande ILNK donne le contrôle à l'éditeur de liens EDILE qui, après s'être réinitialisé, lance l'édition de liens à partir de l'unité symbolique BI.

Le programme objet binaire translatable est produit sur l'unité symbolique BØ, la liste des symboles externes sur l'unité symbolique LL et la liste des symboles internes sur l'unité symbolique LØ.

Lorsque l'éditeur de liens trouve la fin du programme «link editable», il rend le contrôle au superviseur.

Exemple:

* BI	HR	Affectation à BI du lecteur rapide
* BØ	HP	Affectation à BØ du perforateur rapide
* LØ	LP	Affectation à LØ de l'imprimante rapide
* LL	TS	Affectation à LL de la feuille télétype
* <u>ILNK</u>		Activation de EDILE avec demande de réinitialisation
*		Retour à BØS

L'association à l'unité symbolique LØ de ZE permet la suppression de la liste des symboles internes.

Exemple:

```
* BI      HR
* BØ     HP
* LØ     ZE
* LL     TS
* ILNK
*
```

## 2.2.2 – COMMANDE CLNK

La commande CLNK permet de redonner le contrôle à l'éditeur de liens sans réinitialisation.

Exemple 1:

Trois programmes «link editables» sont issus de l'assembleur ASM.

```
* ILNK }      Traitement du programme 1
----- }
* CLNK }      Traitement du programme 2
----- }
* CLNK }      Traitement du programme 3
----- }
*          }
```

Exemple 2:

Il s'agit de générer un programme binaire translatable à partir de deux programmes, l'un issu de l'assembleur ASM, l'autre du compilateur PL 16.

```
* ILNK }      Traitement du programme ASM
----- }
* CLNK }      Traitement de la table d'implantation des sections du programme PL 16
----- }
* CLNK }      Traitement du programme PL 16
----- }
*          }
```

L'ordre programme ASM-programme PL 16 aurait aussi bien pu être inversé.

## 2.2.3 – COMMANDE RLNK

Après chaque traitement l'éditeur de liens signale s'il existe des références non résolues (voir paragraphe 2.3) avant de rendre le contrôle au superviseur BØS.

Lorsque c'est le cas, la commande RLNK permet de demander l'édition de leur liste.

Exemple:

```
* ILNK           Activation de l'éditeur de liens
ERL 09           Indication de l'existence de références non résolues
* RLNK           Demande de l'édition de leur liste
ADRES1          } Liste des références non résolues
PROG2           }
*
```

#### 2.2.4 – COMMANDE ELNK

La commande ELNK est utilisée pour indiquer la fin de l'édition de liens.

Elle donne lieu à la génération dans le programme objet d'un certain nombre d'informations utiles au chargeur ainsi que du checksum du programme qui sera recalculé et vérifié au moment du chargement en mémoire.

La commande ELNK est donc obligatoire et doit terminer toute édition de liens.

Exemple:

```
* ILNK          } Traitement du programme 1
-----
* CLNK          } Traitement du programme 2
-----
* ELNK          Fin de l'édition de liens
*
```

Remarques :

- Lorsqu'il subsiste encore des références qui n'ont pas été résolues (message ERL 09 imprimé à la fin du traitement du dernier programme) l'éditeur de liens force à la valeur zéro les adresses qui correspondent aux symboles externes non définis.
- La commande CLNK est refusée par l'éditeur de liens (message ERL 08) lorsqu'elle intervient après une commande ELNK. Il convient en effet de réinitialiser EDILE (commande ILNK) avant d'effectuer un nouveau traitement.

#### 2.2.5 – COMMANDE TLNK

Lorsque le support utilisé pour le programme objet est le ruban papier, il peut être intéressant d'avoir le programme en plusieurs bandes.

La commande TLNK est alors utilisée pour indiquer la fin de chaque bande intermédiaire, la commande ELNK devant indiquer la fin de la dernière bande.

Exemple:

```
* ILNK          } Traitement du programme 1
-----
* CLNK          } Traitement du programme 2
-----
* TLNK          Fin de la bande objet 1
* CLNK          } Traitement du programme 3
-----
* ELNK          Fin de la bande objet 2 (fin de l'édition de liens)
*
```

**Remarque :**

Les bandes objet issues de l'éditeur de liens doivent être présentées au chargeur translateur dans leur ordre de sortie. Dans l'exemple précédent la bande objet 1 devra être chargée avant la bande objet 2.

**2.2.6 – COMMANDE DLNK**

La commande DLNK permet à l'utilisateur de définir un ou plusieurs symboles externes lorsque les définitions ne sont pas apparues dans les programmes traités jusque là par l'éditeur de liens (à la suite d'un oubli lors de l'écriture des programmes symboliques).

Lorsque EDILE est activé par l'intermédiaire de cette commande, il se met en attente d'une commande de définition dont la forme est la suivante :

< symbole > < adresse hexadécimale > (CR)

Une commande de définition est constituée d'un enregistrement lu sur le périphérique associé à l'unité symbolique Processor Command (PC). Elle se termine par un retour chariot (CR).

**Exemple :**

EXTER1 '75A

Les conventions suivantes sont adoptées par l'éditeur de liens :

- les caractères "espace" figurant dans l'enregistrement sont sans signification
- le caractère "flèche à gauche" ← annule le début de l'enregistrement
- le caractère "flèche en haut" ↑ annule le caractère significatif qui le précède
- lorsque la fin de la phase de définition lui est indiquée par un enregistrement vide (ne comportant que (CR) ), EDILE redonne le contrôle au superviseur B0S.

Lors d'une définition d'un symbole, ce dernier est supposé appartenir au dernier programme traité par EDILE. L'adresse spécifiée dans la commande est donc relative à l'origine de ce programme.

EDILE actualise cette adresse en lui faisant subir la translation introduite en 1.3.1; il imprime ensuite le résultat sur le périphérique associé à l'unité symbolique EL.

Les erreurs suivantes sont détectées par EDILE dans les commandes de définition et donnent lieu à l'impression du message ERL 12 :

- symbole comportant plus de 6 caractères
- symbole n'appartenant pas à la table gérée par l'éditeur de liens
- adresse non hexadécimale
- adresse comportant plus de 4 chiffres hexadécimaux
- adresse actualisée supérieure à 32K.

**2.2.7 – EXEMPLE D'UNE ÉDITION DE LIENS.**

Dans l'exemple suivant les caractères (\*) sont imprimés par le superviseur B0S.

Les messages émis par l'éditeur de liens sont soulignés.

* B1 HR	}	Affectations d'unités fonctionnelles aux unités symboliques
* B0 HP		
* L0 LP		
* ILNK		Lancement de l'édition de liens
<u>S0US-PR0GRAMME</u>		Identification du programme
<u>009A RAC2</u>		Définition de RAC2 en '009A
<u>0231 PARAM</u>		Définition de PARAM en '0231
<u>ERL 09</u>		Existence de symboles non définis
* RLNK	}	Impression des symboles non définis
<u>RESUL1</u>		
<u>RESUL2</u>		
* CLNK		Continuation de l'édition de liens
<u>PR0GRAMME PRINCIPAL</u>		Identification du programme
<u>04B2 RESUL1</u>		Définition de RESUL1 en '04B2
<u>ERL 09</u>		Existence de symboles non définis
* RLNK	}	Impression des symboles non définis
<u>RESUL2</u>		
* DLNK	}	Définition de RESUL2 en '1A4 dans le programme principal
<u>RESUL2 '1A4</u>		
<u>07C2 RESUL2</u>		Définition de RESUL2 en '07C2. (adresse actualisée)
		Fin de la phase de définition (retour chariot)
* ELNK		Fin de l'édition de liens.

Les identifications des programmes traités par l'éditeur de liens sont obtenues par l'intermédiaire de la directive d'assemblage IDP.

### 2.3 – MESSAGES D'ERREUR

Lorsqu'une erreur est détectée, l'éditeur de liens la signale en imprimant sur le périphérique associé à l'unité symbolique Error Listing (EL) un message du type :

ERL xx

xx étant deux chiffres constituant le numéro de l'erreur.

Une erreur peut être fatale ou non.

Dans le premier cas EDILE interrompt le traitement en cours et donne le contrôle au superviseur B0S.

- ERL 00 : Erreur de parité
- ERL 01 : Le checksum de la bande source et celui qui a été calculé par EDILE ne correspondent pas
- ERL 02 : Type du programme source incorrect
- ERL 03 : Erreur dans le programme source
- ERL 04 : Saturation de la table de l'éditeur de liens
- ERL 05 : Passage du compteur de l'éditeur de liens au delà de 32K
- ERL 06 : Adresse translatable actualisée supérieure à 32K
- ERL 07 : Dans le cas des programmes issus du compilateur PL16 l'enchaînement<sup>n</sup> table d'implantation des sections-programme link editable" est incorrect
- ERL 08 : Commande CLNK après une commande ELNK.

Par contre dans le second cas le message d'erreur doit être interprété comme un avertissement.

ERL 09 : Existence de références non résolues

ERL 10 : Multiples adresses de lancement (la dernière est prise en compte)

ERL 11 : Symbole externe déjà défini (la première définition est prise en compte)

ERL 12 : Commande de définition incorrecte

Remarque :

Lorsque sur une commande ELNK il existe des références encore non résolues, l'éditeur de liens ne transmet aucune adresse de lancement dans le programme objet.

## 2.4 – INTERRUPTION D'UNE ÉDITION DE LIENS

L'utilisateur dispose sur le téléimprimeur d'un bouton "appel" qui peut être utilisé pour interrompre une édition de liens en cours et redonner le contrôle au superviseur BØS.

L'éditeur de liens teste les appels en des points où il peut être interrompu sans risque, ce qui permet de demander ultérieurement la continuation du traitement au moyen de la commande CLNK.

Exemple :

Après avoir activé EDILE l'utilisateur se rend compte qu'il a omis d'associer ZE à l'unité symbolique LØ. Il interrompt alors le traitement en cours et revient sous le contrôle de BØS d'où, après avoir réalisé l'association, il relance EDILE.

\* ILNK

ERB 14            Message émis par BØS et du à l'appel opérateur

\* LØ ZE           Affection

\* CLNK            Relance de l'éditeur de liens

-----

## 2.5 – CHARGEMENT DE L'ÉDITEUR DE LIENS

L'éditeur de liens EDILE se présente sous la forme d'une bande binaire translatable.

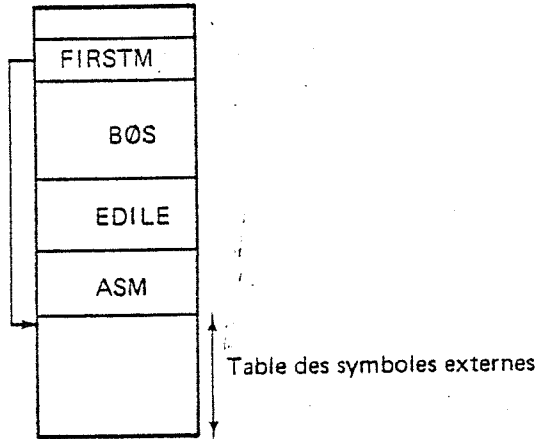
Il s'exécute sous le contrôle du superviseur BØS.

La séquence de programme qui est activée lors du lancement automatique par le chargeur se termine par une requête SVC CAMØ transmettant à BØS les commandes introduites en 2.2.

Il est donc nécessaire, au moment du chargement de EDILE, que le superviseur BØS soit déjà en mémoire.

D'autre part, la table des symboles externes qui est gérée par l'éditeur de liens est limitée inférieurement par le contenu de la mémoire FIRSTM au moment de l'activation de EDILE par la commande ILNK.

Cette mémoire est gérée par le chargeur traducteur ce qui interdit à la table des symboles de venir recouvrir un processeur chargé postérieurement à EDILE.

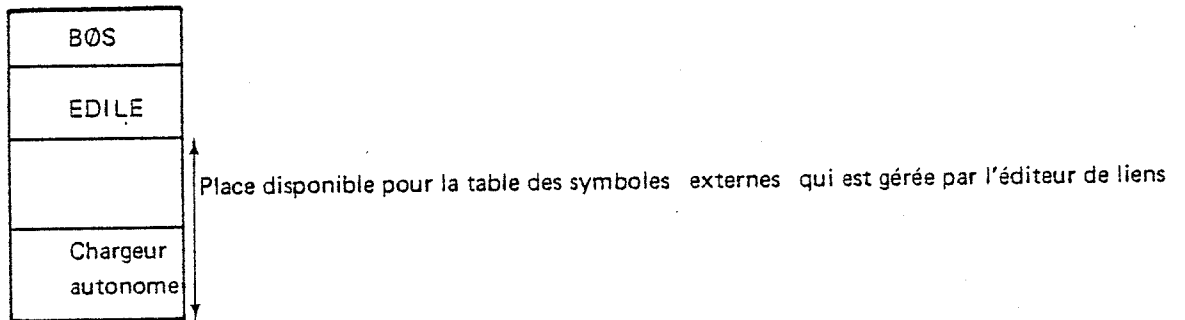


Remarque :

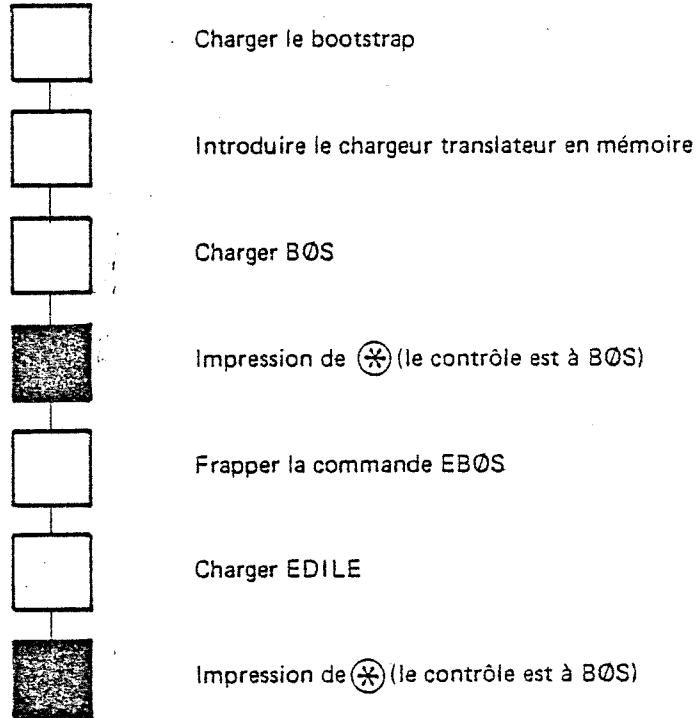
Les implantations des paragraphes suivants ne sont données qu'à titre d'exemples. Pour plus de précisions il est recommandé de se reporter au manuel de référence de BOS

### 2.5.1 – UTILISATION DU CHARGEUR AUTONOME

Une implantation peut être la suivante :



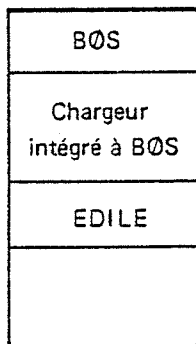
En correspondance la suite des opérations à effectuer est :



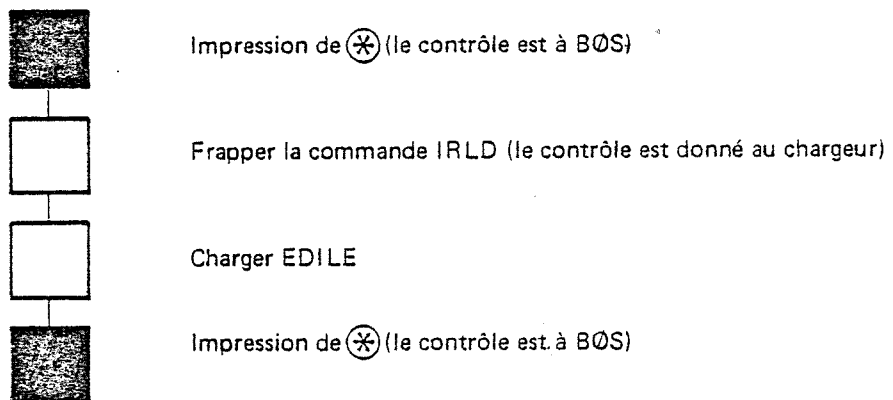
L'éditeur de liens est alors prêt à accepter les commandes de l'utilisateur.

### 2.5.2 – UTILISATION DU CHARGEUR INTÉGRÉ A BØS

Lorsque le superviseur BØS ainsi que le chargeur translateur qui lui est intégré sont en mémoire, une implantation peut être la suivante :



La suite des opérations à effectuer devient :





## ANNEXE : LISTE DES NUMÉROS D'ERREUR

### Erreurs fatales

00	Erreur de parité
01	Erreur de checksum
02	Type du programme source incorrect
03	Erreur dans le programme source
04	Saturation de la table de l'éditeur de liens
05	Passage du compteur de l'éditeur de liens au delà de 32K
06	Adresse calculée supérieure à 32 K
07	Enchaînement "table d'implantation des sections-programme link editable" incorrect
08	Commande CLNK après une commande ELNK

### Avertissements

09	Existence de références non résolues
10	Multiplés adresses de lancement (la dernière est prise en compte)
11	Symbole externe déjà défini (la première définition est seule prise en compte)
12	Commande de définition incorrecte



## FORMULAIRE DE REMARQUES

Soucieux de vous donner satisfaction nous souhaitons recueillir toutes les remarques permettant d'améliorer nos produits.

Nous vous remercions par avance de votre collaboration

Nom : ..... Téléphone : .....

Société : ..... Poste : .....

Adresse : .....

N° d'Affaire : .....

(à relever dans dossier personnalisé)

MODULES CONCERNÉS'

### RÉFÉRENCES

VU II IF ID IE Folio

	VU	II	IF	ID	IE	Folio
.....						
.....						
.....						
.....						

### DÉFAUT CONSTATÉ (ÉVENTUELLEMENT SOLUTION PRÉCONISÉE)

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

..... NOM DE L'ÉMETTEUR  CLIENT Mr. : .....

.....  INTERVENANT Sems Mr. : .....

Date d'établissement .....

Signature .....

Pièces jointes  oui  non

ADRESSE D'EXPÉDITION

SEMS

✓ Promotion des ventes

Service Documentation

Rue de Provence - B.P. N° 58

38130 - ÉCHIROLLES