

C- INFORMATIONS SPECIFIQUES

CONFIGURATION DE NR3

S O M M A I R E

Configuration de NR3

Structure du fichier de configuration

Répartiteur

La primitive SYSTEM de SYSFIC

Composition du système

Contraintes pour l'écriture d'un nouveau module

Points d'entrée dans le système depuis le central

Chargement de langages ou d'applications depuis le central

Algorithme d'initialisation des postes

Les chargeurs standards

Initialisation des postes dans le NANORESEAU (DKBOOT)

En-tête des fichiers NANORESEAU

CONFIGURATION DE NR3

Le chargement du logiciel de gestion du NANORESEAU au central se fait en tapant la ligne de commande :

NR3 (<nom-de-fichier>)

Le paramètre <nom-de-fichier> est le fichier de configuration du système NANORESEAU et est optionnel. S'il est absent, le fichier de configuration utilisé est le fichier NR3.DAT.

Exemples :

| | |
|------------|--|
| NR3 TOTO | Le fichier de configuration est TOTO.DAT |
| NR3 Z.ABC | Le fichier de configuration est Z.ABC |
| NR3 | Le fichier de configuration est NR3.DAT |
| NR3 A:TRUC | Le fichier de configuration est A:TRUC.DAT |

STRUCTURE DU FICHIER DE CONFIGURATION NR3.DAT

| | | | |
|---------|----|------|---|
| VERSION | 0 | (2) | Numéro de version et de sous-version. Il est celui de la version de NR3 correspondante. Par exemple, la version de NR3 de mai 1985 porte le numéro 3.1 |
| NRSYST | 2 | (9) | Nom du fichier chargé par la primitive system de SYSFIC (disque + nom ; pas d'extension) |
| IMPACT | 11 | (1) | Indicateur d'activité de l'imprimante 1 |
| | 12 | (1) | Indicateur d'activité de l'imprimante 2 |
| | 13 | (1) | Indicateur d'activité de l'imprimante 3 |
| | 14 | (1) | Indicateur d'activité de l'imprimante 4 |
| XLATDSK | 15 | (10) | Table de correspondance entre les disques vus par les postes et les volumes vus au central. Si la table commence par 1, 3, 0,... les postes verront comme disque A: le disque A: du central et comme disque B: le disque C: du central. Avec les bits 6 et 7 ayant la signification suivante : bit 7 = 0 bit 6 = 0 floppy bit 7 = 0 bit 6 = 1 Ram disk bit 7 = 1 bit 6 = 0 Hard disk |
| DSKIMP | 25 | (1) | Disque (vu du poste) où envoyer les fichiers listings. |
| OCTSYS | 26 | (1) | Octet réservé au système. |
| IDCAR | 27 | (1) | Nombre d'identifications cartouches soit n*16 n fois... |
| | 0 | (32) | Les octets d'identification de cartouche |
| | 32 | (12) | Le nom du fichier complément de langage associé à cette cartouche (disque + nom + extension) |

Le source du fichier NR3DAT.ASM sera fourni aux développeurs. Ce fichier correspond au fichier NR3.DAT figurant sur la disquette standard. Les développeurs pourront fabriquer d'autres fichiers de configuration. Le fichier de commandes NR3DAT.BAT permet d'enchaîner automatiquement les différentes étapes de la création de NR3.DAT à partir de NR3DAT.ASM.

REPARTITEUR

Pour répartir les requêtes réseau entre les différents modules au central, le module RT utilise une table structurée ainsi :

```
TABMOD      0      (2)  Nombre de modules soit n
              n fois...
              0      (1)  Le code tâche réseau
              1      (2)  L'adresse du module correspondant
```

Exemple :

```
-----
TABMOD:      DW      (OFFSET FTACHES-OFFSET TACHES)/ 3 ; NOMBRE DE TACHES
TACHES:

DB          0
DW          OFFSET DEBOOT

DB          9
DW          OFFSET SYSFIC

DB          11
DW          OFFSET SPOOL
```

FTACHES LABEL NEAR

L'étiquette TABMOD est publique. Le source de cette table est fourni aux développeurs qui doivent le modifier en fonction de leurs besoins, l'assembler et l'inclure dans l'édition de liens finale.

LA PRIMITIVE SYSTEM DE SYSFIC

Lorsqu'un poste fait appel à cette primitive, le central nettoie l'espace de travail du poste au central (voir la primitive CLEAR) et télécharge dans la mémoire du poste le module système des postes.

Le nom de ce module est dans le fichier de configuration ; c'est NRDOS dans la version standard. Il existe en fait deux versions du module système d'extensions .M05 et .T07 selon la machine visée.

Le module système permet de télécharger les différents langages ou applications en langage machine disponibles au central. On a le schéma suivant :

| Langages | | | Module système (exemple NRDOS) |
|---|------|------|-----------------------------------|
| BASIC SYSTEM | ---> | <--- | A>BASIC |
| LOGO SYSTEME | ---> | <--- | A>LOGO |
| LSE FIN | ---> | <--- | A>LSE |
| Application en langage machine (ex: APPLI) Commande de sortie | ---> | <--- | A>APPLI |

Par ailleurs, lors de l'allumage d'un poste dans le réseau, le mécanisme d'initialisation de ce poste (DKBOOT) conduit le central à télécharger dans le poste un module binaire qui se trouve dans un fichier. Le nom de ce fichier se trouve également dans le fichier de configuration. On peut le modifier et le remplacer par un module de chargement spécifique (par exemple, le module système). Cette structure procure une grande souplesse pour réaliser des applications intégrées fonctionnant sur le NANORESEAU.

COMPOSITION DU SYSTEME

Le système de gestion du NANORESEAU au central est livré sous deux formes différentes :

- Un programme exécutable NR3.EXE sous MSDOS qui constitue la version standard,
- Un ensemble de modules relogeables dans le format de l'assembleur MASM de Microsoft sous MSDOS :

| | |
|--------------|-----------------------------------|
| CTxxxxxx.OBJ | Couches de transport |
| SYSFIC.OBJ | Système de fichiers du NANORESEAU |
| SPOOL.OBJ | Gestion de l'imprimante |
| IU.OBJ | Interface utilisateur au central |
| RT.OBJ | Répartiteur des tâches au central |
| TABMOD.OBJ | Table des modules utilisée par RT |
| INI.OBJ | Configurateur du système |

Ces différents modules contiennent des points d'entrée (directive "PUBLIC" de MASM) et des références externes (directive "EXTRN" de MASM).

L'ensemble de ces modules donne le programme NR3.EXE par édition de liens.

Le nom CTxxxxxx.OBJ doit être compris de la façon suivante :

| | |
|-------------|--|
| CTZ16.OBJ | Pour le SIL'2 16 de LEANORD |
| CTBM30.OBJ | Pour le BM30 de BULL MICRAL |
| CTP1600.OBJ | Pour le PERSONNA 1600 de LOGABAX |
| CTMAX20.OBJ | Pour le MAX 20 de MATRA MICRO SYSTEMES |
| CTG3.OBJ | Pour le GOUPIL 3 de SMT |
| CTIBM.OBJ | Pour le PC de IBM |

Cet ensemble de modules, muni de sa documentation, permettra à un développeur de fabriquer des applications spécifiques pour le NANORESEAU en incorporant et/ou remplaçant un ou plusieurs modules.

CONTRAINTES POUR L'ECRITURE D'UN NOUVEAU MODULE

- Etre écrit dans le format de l'assembleur MASM

- Utiliser trois segments :

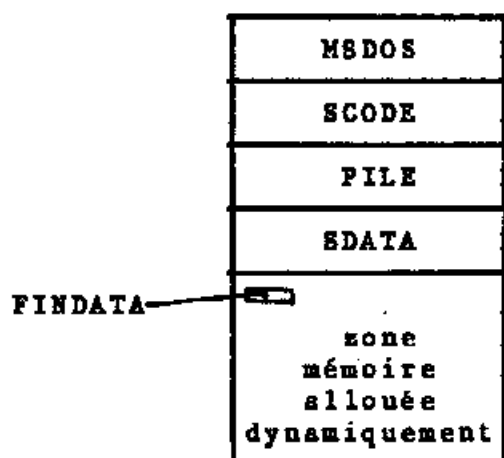
| | | |
|-------|----------------|-------------------|
| SCODE | segment public | ;pour le code |
| SDATA | segment public | ;pour les données |
| PILE | segment stack | ;pour la pile |

- Les appels entre les différents modules se font par des `call near`.

- Les sous-programmes publics des différents modules doivent conserver aux registres de segments les valeurs suivantes :

```
CS = SCODE
DS = ES = SDATA
SS = PILE
```

- L'occupation mémoire doit être la suivante :



Le symbole public "FINDATA" référence le premier octet de SDATA libre. Il est utilisé par le système pour gérer le reste de la mémoire (zone des buffers).

Limitation :

```
64 K de code au maximum
64 K de données maximum (hors buffers)
64 K de pile au maximum
```

POINTS D'ENTREE DANS LE SYSTEME DEPUIS LE CENTRAL

Un module au central dispose de deux sortes de points d'entrée dans le système au central :

Les sous-programmes de SYSFIC

Ce sont les mêmes primitives que celles qu'utilisent les postes et qui sont documentées dans le chapitre "Primitives fichiers du NANORESEAU" (OUVFL, CREFI, CATP, etc... par exemple). Le mode d'appel à ces primitives est le suivant :

- Mettre le booléen RESEAU? à faux (00).
 - Remplir la zone consigne ZCON conformément à la documentation de la primitive invoquée.
- | | | | |
|-----------|-----|--------------------------------|---|
| ZCON + 0 | (1) | Longueur de la consigne | <60 |
| ZCON + 1 | (1) | Expéditeur | 1 à 31 postes du réseau 32 à 255 modules au central 0 numéro interdit |
| ZCON + 2 | (1) | Code tâche réseau : | SYSFIC = 9 |
| ZCON + 3 | (1) | Code tâche application. | C'est le code de la primitive invoquée (OUVFL=36, LIRFI=40, par exemple) |
| ZCON + 4 | (2) | Nombre d'octets | (poids fort, poids faible) |
| ZCON + 6 | (1) | Page | |
| ZCON + 7 | (2) | Adresse | (poids fort, poids faible) |
| ZCON + 9 | (1) | Type d'ordinateur | (0=M05, 1=T07, 2=T07 70) |
| ZCON + 10 | (1) | Application | (1=BASIC, 2=LOGO, 3=LSE, etc...) |
| ZCON + 11 | (?) | Les paramètres supplémentaires | de l'appel |
- Faire un call near à SYSFIC
 - Remettre le booléen RESEAU? à vrai (0ffh)
 - Les informations retournées par SYSFIC se trouvent dans le buffer de compte rendu à l'adresse CRDU.

NOTES

- Il est impératif d'utiliser les primitives de SYSFIC concernant les fichiers et de ne pas passer directement par MSDOS pour ne pas perturber la logique de partage des fichiers entre postes et pour utiliser correctement le mécanisme de mémoire cache utilisé par le système.

- La page (ZCON + 6) et l'adresse (ZCON + 7) doivent être compris de la façon suivante :

Si le bit 7 de la page est nul, l'adresse est l'offset de la zone mémoire dans SDATA,

Si le bit 7 de la page est non nul, l'adresse de la zone de données est calculée sur 20 bits en prenant les 4 bits de poids faible de la page et les 16 bits de l'adresse :

| | |
|-----------------|----------------------------------|
| Page | 1... aaaa |
| Adresse 16 bits | bbbb cccc dddd eeee |
| Adresse finale | aaaa bbbb cccc dddd eeee |

- Les seules références publiques à utiliser sont :

SYSFIC : near (dans SCODE)
 ZCON : byte (dans SDATA)
 CRDU : byte (dans SDATA)
 RESEAU?: byte (dans SDATA)

Les autres sous-programmes

Ils sont appelés simplement par des call near. Les paramètres sont passés par des registres et/ou des zones mémoire conformément à leur documentation.

CHARGEMENT DE LANGAGES OU D'APPLICATIONS DEPUIS LE CENTRAL

Il s'agit de télécharger dans la mémoire du poste des plages de code 6809 et d'en lancer l'exécution. Cette opération est facile à réaliser : il suffit de faire appel à CHBIN.

Cependant, tous les fichiers binaires qui se trouvent au central ne sont pas bons à charger ; on ne peut charger, sans risque de "planter" les postes, que des langages ou applications en langage machine prévus pour un tel chargement. Ces programmes binaires doivent en particulier prendre en charge l'initialisation du poste à leur lancement.

C'est pourquoi le téléchargement d'un langage ou d'une application binaire dans un poste doit se faire de la façon suivante (voir la documentation sur les formats binaires) :

- Ouvrir le fichier correspondant
- Vérifier que le type est 2 (langage machine)
- Vérifier que le mode est 0 (mode binaire)
- Lire les 6 premiers octets du fichier
- Vérifier que le premier octet est égal à 1 (signature du fichier)
- Vérifier le type de machine (2e octet)
- Vérifier l'octet de chargement (3e octet)
- Mettre le poste en attente
- Faire appel à CHBIN en se faisant passer pour le poste

ALGORITHME D'INITIALISATION DES POSTES

A- LE POSTE N'EST PAS EQUIPE D'UNE CARTOUCHE de ROM :

Mise sous tension de l'appareil

Initialisation du poste par le MONITEUR

Initialisation du langage

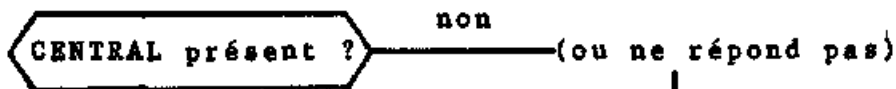
Le langage doit faire un DEBOOT

L'EPRom réseau initialise le poste dans le réseau, le circuit de communication.

Elle positionne les pointeurs TRCT, CTRLC, CRDU sur une demande d'émission.

Elle initialise ORD (0=TO7, 1=MO5, 2=TO7-70)

Puis fait appel au SERVEUR en envoyant sa SIGNATURE de 32 octets d'identification.



non

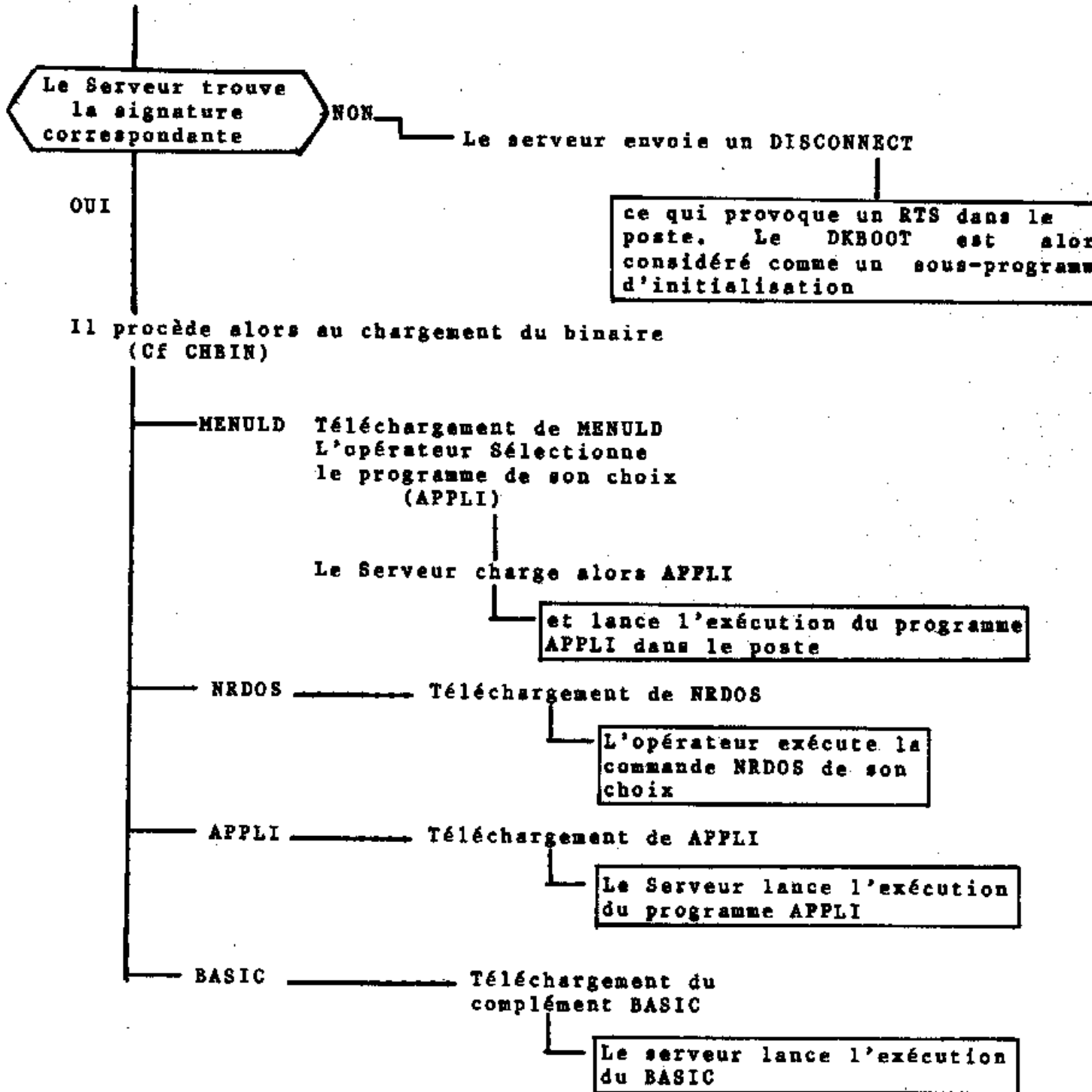
(ou ne répond pas)

Dans ce cas, l'EPRom réseau provoque un départ à froid du langage en exécutant la séquence:
 CLR DKFLG = 2080H (MO5)
 6080H (TO7)
 JMP (COLD) = 0EFFFH (MO5)
 0001EH (TO7)

Le SERVEUR reçoit cet appel

Le Serveur initialise les paramètres internes liés à ce poste (Identification Std, il ferme tous les fichiers ouverts par le poste (les fichiers listings ouverts en les envoyant au "SPOOL"). Il relâche tous les fichiers réservés par ce poste (Cf CLEAR).

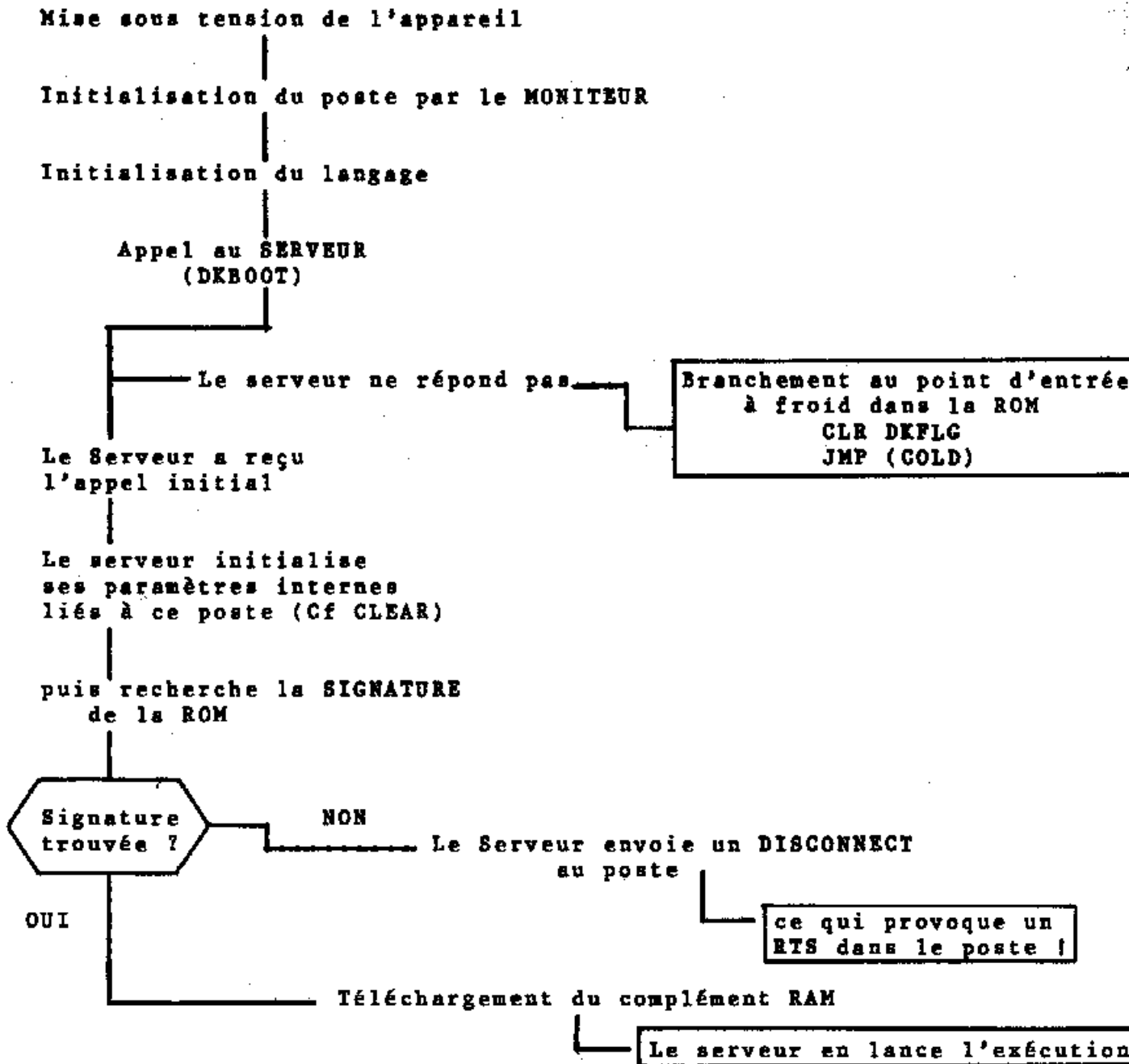
Puis le Serveur recherche la signature du poste dans la table IDCAR.



Si le chargement échoue, le Serveur affiche un message d'erreur sur l'écran du poste et télécharge le module système NRDOS.

- Les causes d'erreur peuvent provenir :
- d'un fichier inexistant,
 - d'un défaut de structure du fichier,
 - de l'absence de cartouche RAM...

B- LE POSTE EST EQUIPE D'UNE CARTOUCHE de ROM



REMARQUES

Il ne faut pas accorder plus d'importance qu'elle ne mérite à cette table de correspondance entre des signatures de cartouches et des modules binaires à télécharger. En fait, elle ne devrait contenir que les signatures des cartouches de langages ou d'applications qui existaient antérieurement au NANORESEAU et qui admettent un complément de langage adaptable.

Le cas exemplaire est celui du BASIC dont la partie traitant les opérations sur fichiers est normalement lue à partir du disque système et qui, dans l'optique du NANORESEAU, a complètement été refaite pour s'adapter au système de fichiers du NANORESEAU.

L'autre cas connu à ce jour est celui de LOGO.

Ces exemples ne sont pas à suivre.

Il est préférable de mettre l'intégralité des langages ou applications à venir soit :

- dans un fichier binaire étendu (fonctionnement prévu en cartouche de RAM),
- dans une ROM.

Examinons le cas de quelqu'un voulant développer une application en langage machine pour le M05 ou le T07 en NANORESEAU.

Deux hypothèses sont à envisager :

- L'application est destinée à fonctionner en ROM

Dans ce cas, il est inutile et compliqué d'avoir recours à un complément de logiciel en RAM téléchargeable depuis le central. La meilleure solution consiste à tout faire tenir dans la cartouche de ROM. La signature de cette application n'a pas à être déclarée au central.

A l'initialisation, l'application devra initialiser le poste dans le réseau en faisant :

```
JSR DKBOOT
<instruction-suivante>
```

Le central, ne connaissant pas la signature de cette application, procédera à un DISCONNECT et le DKBOOT sera alors vu comme un sous-programme : <instruction-suivante> sera exécutée.

L'application devra alors initialiser les pointeurs TRCT, CTRLC et CRDU convenablement. Elle devra également prendre en charge l'octet APPLI (0=application inconnue, 1=BASIC, 2=LOGO, 3=LSE, etc...).

- L'application est destinée à fonctionner en cartouche de RAM

Elle est alors livrée sous la forme d'un fichier binaire au format étendu. C'est le système qui télécharge le code dans le poste (en tenant compte des banques), qui positionne l'octet APPLI et qui lance l'exécution de l'application.

A l'initialisation, l'application peut faire appel à DKBOOT ; ce n'est pas obligatoire dans la mesure où le réseau est déjà initialisé. L'application peut faire appel à la primitive CLEAR pour nettoyer le descripteur du poste au central. De toute façon, l'application devra initialiser les pointeurs TRCT, CTRLC et CRDU convenablement.

LES CHARGEURS STANDARDS

Il existe deux chargeurs dans le NANORESEAU : NRDOS et MENU. Ces deux chargeurs utilisent la même primitive du système de fichier CHBIN qui charge les fichiers binaires (format simple et format étendu) depuis le central. La différence entre ces deux chargeurs est leur interface utilisateur.

NRDOS se présente comme un interpréteur de commandes analogue à celui de MSDOS : il affiche le message d'invite A> et attend une commande de l'utilisateur.

Si cette commande n'est pas résidente, il cherche à lancer l'exécution d'un fichier binaire (d'extension .MO5 ou .TO7 selon les cas).

Pour charger un langage ou une application, il suffit donc de taper son nom sous NRDOS.

MENU présente à l'utilisateur un choix entre plusieurs langages ou applications présentes au central et téléchargeable dans les cartouches de RAM.

Le libellé de ce choix est modifiable par une procédure simple. On peut insérer dans le MENU standard une nouvelle application en donnant le texte qui s'affiche à l'écran du poste lors du choix et le nom du fichier binaire correspondant.

EN-TETE DES FICHIERS NANORESEAU

L'EN-TETE DES FICHIERS NANORESEAU

Les fichiers créés par le NANORESEAU sont des fichiers MSDOS dont les 128 premiers octets forment une en-tête qui contient différentes informations utilisées par le système et les applications.

- +9 (8) Les 8 caractères *NRUSTL* constituent la marque qui permet de distinguer les fichiers NANORESEAU.
- +8 (8) Identification du fichier. C'est l'identification du poste au moment de la création du fichier. Si le poste avait l'identification standard, ces huit caractères sont des espaces (§20).
- +16 (1) Nature du fichier :
 - 1 fichier standard
 - 2 fichier structuré LSE
- +17 (1) Type du fichier (Compatibilité DOS Thomson)
 - 0 programme BASIC (ASCII ou Binaire)
 - 1 données BASIC
 - 2 langage machine
 - 3 fichier source
- +18 (1) Mode du fichier (Compatibilité DOS Thomson)
 - 0 binaire
 - §ff ASCII
- +19 (3) Longueur du fichier pour MSDOS y compris le secteur d'en-tête. Cette longueur est codée sur trois octets avec la convention du 8088/86 : poids faible, poids moyen et poids fort.
- +22 (0) Un octet nul
- +23 (1) Statut du fichier :
 - 0 fichier en lecture et écriture
 - §ff fichier en lecture seulement
- +24 (2) Version et sous-version du logiciel réseau ayant créé ce fichier.
 Exemple : pour les fichiers créés par la version de
 ----- mai 1985, ces deux octets ont les valeurs
 3 et 1.

Suite EN-TETE DES FICHIERS NANORESEAU

- +26 (3) Date de création du fichier. Cette date est codée sous la forme aa/mm/jj
- +29 (3) Date de la dernière modification du fichier. Cette date est codée sous la forme aa/mm/jj
- +32 (1) Ordinateur ayant créé ce fichier :
- 0 T07
 - 1 M05
 - 2 T07 70
- +33 (1) Type du langage ou de l'application ayant créé ce fichier :
- 0 Langage ou application inconnu
 - 1 BASIC 1
 - 2 LOGO
 - 3 LSE
- +34 (46) 46 octets nuls
- +80 (48) Octets réservés aux applications.
Ils sont accessibles par les primitives LIRATT et ECRATT.

IDENTIFICATEURS DES LOGICIELS THOMSON

L'identificateur d'un logiciel THOMSON est constitué des 32 derniers octets de l'espace cartouche qu'il utilise. Sa forme standard est la suivante :

| | | |
|-----|-----------------|--|
| FCC | XXXXXXXX | : Nom du logiciel |
| FCB | 04 | : EOT |
| FCB | 0,0,0,0,0,0,0,0 | : Suite de 00 à ajouter pour compléter le nom à 20 octets |
| FDB | XX | : Checksum sur 2 octets obtenu en sommant tous les octets du nom, y compris EOT et en y ajoutant 5555 Hexa |
| FCB | 0,0,0,0,0,0,0,0 | : 8 octets non utilisés (à 0 par exemple) |
| FDB | DEBUT | : Adresse de début de programme |

Exemple d'identificateur :

| | |
|-----|---|
| FCC | /SCRIPTOR/ |
| FCB | EOT |
| FCB | 0,0 |
| FCB | 57CF |
| FCB | 0,0 |
| FDB | LDEBU |

**PRECAUTIONS A PRENDRE LORS DU REDEMARRAGE A CHAUD
D'UNE APPLICATION NANORESEAU**

Afin de garantir qu'une application, tournant sur un MO5 dans le NANORESEAU, redémarre à chaud proprement, suite à un RESET du poste, il est impératif d'implémenter les fonctions suivantes :

- Vérifier que le numéro du poste dans le réseau est correct,
- Vérifier que le vecteur d'interruption propre au contrôleur de réseau est bien positionné,
- Réinitialiser le contrôleur de réseau.

Pour ce faire, il faut et il suffit d'implanter la routine suivante :

```

INIRES      EQU *
SEA041     LDX  #IRQE      ; Initialise le vecteur d'IT réseau
BF2064     STX  IRQPT
BFA7D8     LDA  NUPOST    ; Initialise le numéro de poste
841F      ANDA #1F        ; Pour utilisation par le logiciel réseau
B72052     STA  NUPO
BDA02B     JSR  RELACH    ; Initialise le contrôleur réseau
;
          IRQE      EQU  $A041
          IRQPT     EQU   2064
          NUPOST    EQU   A7D8
          NUPO      EQU   2052
          RELACH    EQU   A02B
;

```

La routine RELACH est accédée par un point d'entrée vectorisé situé en A02B dans la REPRON réseau. cette adresse est donc garantie, même en cas de modification de cette REPRON.

Cette routine INIRES doit être exécutée le plus tôt possible après le RESET afin que, si le poste est interrompu par un appel extérieur réseau, il soit en mesure de répondre correctement à cet appel.

Il est préférable d'exécuter INIRES avant d'autoriser les IT au niveau du microprocesseur afin de ne pas risquer d'être interrompu par le réseau au moment de cette initialisation. Il ne faut pas oublier que le moniteur rentre dans l'application dans l'état IT masquées (I et F à 1 dans CC). Ce qui fait que l'application n'a pas besoin d'inhiber elle-même les IT. Par contre, elle doit explicitement les autoriser, même si ce n'est pour son propre usage ! En effet, le logiciel exige une IT pour fonctionner.

SPECIFICITES DU BASIC M05 POUR NANORESEAU

S O M M A I R E

- 1- DOCUMENTS DE REFERENCE
- 2- POSSIBILITES SUPPLEMENTAIRES
 - 2-1 Utilisation des imprimantes connectées au serveur
 - 2-2 Manipulations de copies d'écran
 - 2-3 Echanges de données entre postes ou entre poste et serveur
 - 2-4 Recopie d'écran, de programme ou de bloc mémoire
 - 2-5 Date et Heure
 - 2-6 Déclaration de fichiers temporaires
 - 2-7 Envoi de consignes réseau
 - 2-8 Diffusion de programmes

SPOO:

PICTURE PRINT

PICTURE LOAD

PICTURE SAVE

NODE:

INNODE\$

CLONE

TIME\$

DATE\$

TEMP ON/OFF

NET\$

1- DOCUMENTS DE REFERENCE

Les documents de base décrivant le BASIC Microsoft 1.0 des unités centrales MO5 sont regroupés dans deux volumes :

- Le Guide du MO5 chez CEDIC-NATHAN No ISBN : 2-7124-0522-6
- Le Basic DOS du T07/T07-70/MO5
chez CEDIC-NATHAN No ISBN : 2-7124-0584-6

Par rapport à ces documents d'origine qui s'appliquent au BASIC des machines MO5 individuelles munies de leur lecteur de disquette et de la disquette système DOS, les spécifications du BASIC des machines dans le NANORESEAU diffèrent par certains points documentés ci-après. Le BASIC des MO5 en NANORESEAU a en effet été développé pour bénéficier pleinement des possibilités offertes par la présence d'un serveur et d'autres postes avec lesquels on peut communiquer.

2- POSSIBILITES SUPPLEMENTAIRES**2-1 Utilisation des imprimantes connectées au serveur**

Le BASIC prévoit l'impression de listings de programmes ou de résultats sur le spool d'imprimante au central à l'aide du canal d'entrée/sortie "SPOO:"

Deux possibilités sont offertes :

- pour imprimer le contenu d'un programme :

```
LIST "SPOO:((NC,NEX)NI)
```

- pour imprimer des résultats :

```
OPEN"O",N,"SPOO:((NC,NEX,MODE)NI)
```

suivi d'ordres PRINT N et se terminant par un ordre CLOSE N

La différence, par rapport à la syntaxe classique du canal d'imprimante locale "LPRT:", se situe dans le complément d'arguments optionnels pour décrire le spooler du central. Au nombre de colonnes du format déjà décrit habituellement par (NC), sont ajoutés les arguments suivants :

- **NEX** : nombre d'exemplaires du document. Ce nombre est un entier compris entre 1 et 10. Par défaut, en l'absence de paramètre explicite, ce nombre est pris égal à 1.

- **MODE** : type de codage des données envoyées par le poste.

MODE = 0 mode "transparent" où tous les codes envoyés au serveur sont transmis intégralement et sans transcodage pour permettre au poste de piloter directement l'imprimante de son choix

MODE = 1 codes au standard "IBM" directement compatibles avec l'imprimante MANNESMANN-TALLY

MODE = 2 BASIC "M05" correspondant au codage interne du moniteur du M05 : ce mode donne des fichiers imprimables exactement identiques à des fichiers de programmes BASIC sauvegardés en ASCII.

MODE = 3 est un mode propre au langage LSE

MODE = 4 mode "imprimante THOMSON" employé lorsqu'on donne au central un codage compatible avec celui employé par les imprimantes à impact vendues sous la marque THOMSON et habituellement connectées aux machines individuelles

MODE = 16 mode "recopie d'écran" indiquant que les données envoyées représentent un écran graphique compacté selon l'algorithme standard des fichiers .MAP (voir au paragraphe suivant les copies d'écran).

Par défaut, le mode employé est le mode 2. Dans l'instruction LIST, le mode ne peut être spécifié puisque les données envoyées sont nécessairement au mode numéro 2.

- **NI** : numéro de l'imprimante sur laquelle on veut que l'impression s'exécute. Ce numéro est un entier compris entre 0 et 3. Par défaut, en l'absence de paramètre explicite, c'est l'imprimante numéro 0 qui est prise.

Par exemple, l'instruction OPEN "0",3,"SPOO:" ouvre le canal 3 vers l'imprimante 0 pour 1 exemplaire en mode M05.

L'instruction LIST "SPOO:(,4)1" imprime 4 exemplaires du programme résident sur l'imprimante numéro 1.

2-2 Manipulations de copies d'écran

Le BASIC fournit trois instructions pour imprimer, sauvegarder et retrouver une copie d'écran graphique :

- PICTURE PRINT (NI)
- PICTURE SAVE "Nom de fichier"
- PICTURE LOAD "Nom de fichier"

Le fichier créé est au format compacté compatible avec les autres logiciels THOMSON susceptibles de faire du graphisme. Le suffixe employé pour ces fichiers est .MAP.

La compaction permet d'économiser la place sur disque. L'algorithme de compaction/décompaction est inclus dans le BASIC : la compaction est faite à chaque PICTURE PRINT ou PICTURE SAVE.

La décompaction est faite par le PICTURE LOAD.

Lors d'un PICTURE PRINT, un fichier compacté est créé qui est envoyé au spool d'impression après décompactage par le serveur.

L'impression se fera dès que le spool le permettra sur l'imprimante NI en un exemplaire. NI est le numéro optionnel de l'imprimante, c'est un entier compris entre 0 et 3. Par défaut, en l'absence de paramètre explicite, c'est l'imprimante 0 qui est prise.

PICTURE SAVE peut également servir à faire des impressions d'écrans graphiques en faisant par exemple : PICTURE SAVE "SPOO:(0,,16)NI". Dans cette syntaxe, on retrouve les paramètres du canal "SPOO:" : la largeur en nombre de colonnes est spécifiée à une valeur nulle de façon à empêcher les retours de chariot au milieu de l'impression; le nombre d'exemplaires est laissé à 1 (valeur par défaut) le mode est spécifié à 16 afin de faire savoir au serveur qu'il reçoit un fichier compacté de type .MAP qu'il doit décompacter avant de l'imprimer.

2-3 Echanges de données entre postes ou entre poste et serveur

Un mécanisme général est prévu pour que les postes puissent se communiquer des données ou des messages ou envoyer/recevoir des informations vers un périphérique connecté à un autre poste ou au serveur.

Ce mécanisme est conforme aux entrées/sorties générales du BASIC et fonctionne selon le principe suivant :

- un poste désirant envoyer des données vers un autre poste doit ouvrir un canal d'émission vers le poste destinataire. Afin que le destinataire puisse recevoir les informations envoyées par l'émetteur, il doit lui-même ouvrir un canal de réception en provenance de l'émetteur. Ceci s'accomplit à l'aide du mécanisme d'entrées/sorties classique du BASIC : une instruction OPEN "O", N"...." ouvre un canal en émission tandis que OPEN "I", N, "..." ouvre un canal en réception (voir syntaxe complète à la fin du paragraphe). Afin de ne pas générer d'erreur sur le poste émetteur au moment de l'ouverture du canal en sortie, il est recommandé d'ouvrir auparavant le canal de réception sur le poste récepteur. En effet, une ouverture en émission vers un correspondant encore inexistant génère une erreur Not Open alors qu'une ouverture en réception vers un émetteur encore inexistant ne provoque pas d'erreur.
- une fois les canaux de transmission établis, les émissions et réceptions de données se font à l'aide des instructions classiques du BASIC :

PRINT #N et PRINT USING #N, pour émettre
 INPUT #N, INPUT\$(I,#N) et LINE INPUT #N, pour recevoir

Les instructions à base de INPUT du BASIC restent bloquées s'il n'y a pas de caractère disponible en provenance de l'émetteur, une nouvelle instruction de lecture non bloquante a été rajoutée :

INNODE\$(I,#N) qui reçoit I caractères en provenance de l'émetteur : elle se termine dès que le nombre I est atteint ou qu'il n'y a plus de caractères disponibles

- une fois les échanges achevés, pour fermer les canaux de transmissions, on utilise l'instruction standard CLOSE N

La syntaxe exacte pour ouvrir les canaux est la suivante

OPEN "O",N,"NODE: (chaîne de caractères) Np"
 OPEN "I".....

- N est le numéro de canal d'entrée/sortie
- NODE signifie qu'on s'adresse à un noeud du Réseau

- la chaîne de caractères entre parenthèses est optionnelle : elle sert à décrire, quand on s'adresse au central, un périphérique connecté à celui-ci auquel on veut s'adresser depuis un poste.
- Np est le numéro de poste dans le réseau, compris entre 0 et 31, 0 étant le numéro du serveur.

Par exemple, OPEN "I",1,"NODE:8" ouvre un canal en lecture vers le poste numéro 8.

OPEN "O",1,"NODE:(RS232,2400,2S,NP)O" ouvre un canal en écriture vers une liaison d'entrée/sortie du serveur décrite par la chaîne de caractères RS232, 2400, 2S, NP qui pourrait être, par exemple, une liaison RS232. Cela suppose évidemment que le serveur possède une telle interface et dispose du logiciel pour la piloter depuis un poste du réseau.

2-4 Recopie d'écran, de programme ou de bloc mémoire d'un poste à l'autre

Il est possible de recopier dans un poste l'écran ou un programme BASIC ou un bloc de mémoire en provenance d'un autre poste du réseau. Cinq options sont données :

- CLONE N : recopie le programme BASIC du poste N. Cette instruction fonctionne comme si on "photographiait" le programme du poste N et qu'on l'envoyait tel quel au poste demandeur. Si le poste N est en cours d'édition de son programme, aucune garantie n'est donnée quant à la validité du programme reçu par le demandeur.
- CLONE N,R : recopie le programme BASIC du poste N et lance son exécution. Comme précédemment, aucune garantie ne peut être donnée de la validité du programme si le poste N est en édition.
- CLONE N,C : concatène, au programme déjà existant dans le poste, le programme BASIC du poste N. Cette concaténation suppose que les numéros de lignes des deux programmes ainsi regroupés sont disjoints et cohérents afin d'obtenir un programme unique. Aucune vérification n'est faite quant à la cohérence des numéros de lignes.

- CLONE N,S : recopie l'écran du poste N. Cette instruction fonctionne aussi comme une photographie de l'écran du poste N qui est envoyée au poste demandeur.
- CLONE N,M,AD,AF(AC) : prend le bloc mémoire situé entre AD (adresse de début) et AF (adresse de fin) dans le poste N et le recopie à partir de l'adresse optionnelle AC (adresse de chargement) dans le poste demandeur. Si l'adresse AC n'est pas précisée dans l'instruction, alors c'est l'adresse de début de bloc AD qui est utilisée comme adresse de chargement.

Par exemple :

- CLONE 5 : recopie le programme BASIC du poste 5
- CLONE 3,S : recopie l'écran du poste 3
- CLONE 3,S : recopie et exécute le programme BASIC du poste 3
- CLONE 11,M, H7200, H79FF, H6500 : recopie à partir de l'adresse H6500 le contenu du bloc de mémoire situé entre H7200 et H79FF du poste 11.

2-5 Date et Heure

Il est possible de connaître la date et l'heure telles qu'elles sont données par la tête du réseau. Deux variables sont réservées à cet effet :

- DATE\$: c'est une chaîne de caractères de la forme JJ-MM-AAAA, par exemple PRINT DATE\$ renvoie le 12-04-1985.
- TIME\$: c'est une chaîne de caractères de la forme HH:MM:SS, par exemple PRINT TIME\$ renvoie 15:23:40

2-6 Déclaration de fichiers temporaires

Il est possible à un poste de déclarer au serveur que les fichiers qu'il va créer sont de type temporaire, c'est-à-dire que ces fichiers pourront être détruits par le serveur à la fin de la session. Ceci permet, par exemple, de ne pas encombrer inutilement les disques du serveur avec des fichiers d'exercices que l'on ne souhaite pas conserver.

L'instruction de déclaration est : TEMP ON

Tous les fichiers créés après cette déclaration sont marqués comme temporaires pour le serveur.

Pour revenir au mode de création normal, on fait : TEMP OFF

2-7 Envoi de consignes réseau

Les programmeurs désireux d'effectuer un transfert de données sur le réseau en descendant au niveau du protocole peuvent écrire eux-mêmes leurs consignes et traiter les comptes rendus à l'aide de l'instruction NET\$.

Par exemple B\$ = NET\$(CONSIG\$) envoie la consigne contenue dans la chaîne de caractères CONSIG\$ et renvoie le compte rendu (s'il y en a un) afférent à la consigne dans la chaîne de caractères B\$.

Il est IMPERATIF de se conformer à la structure des consignes telle qu'elle est spécifiée dans la documentation à l'usage des développeurs du NANORESEAU. Tout envoi de consigne non-standard ou ne figurant pas dans la liste des consignes valides peut entraîner un arrêt complet de tout le réseau et peut détruire le contenu des fichiers au serveur.

2-8 Diffusion de programmes

Il est possible, depuis le serveur de réseau, de diffuser aux postes travaillant en BASIC ou à certains d'entre eux un programme BASIC.

Cette facilité n'est accordée qu'au serveur : se reporter à la documentation correspondante.

Un poste ayant reçu un programme diffusé depuis le serveur perd le contexte dans lequel il se trouvait avant la diffusion. Le programme reçu s'exécute automatiquement (comme si le poste avait fait RUN). Le lancement de l'exécution de ce nouveau programme ferme automatiquement tous les canaux d'entrée/sortie qui pouvaient être ouverts au moment de la diffusion.

2-9 SYSTEM

L'instruction **SYSTEM** permet de quitter le BASIC et de revenir au niveau système en exécutant le programme **NRSYST** qui lui est associé et est défini dans le fichier **NR3.DAT**.

Celui-ci peut être :

- soit **MENU**,
- soit **NRDOS**,
- soit tout autre fichier auto-exécutable dans le poste.

Il faut s'assurer que ce programme est en mesure de réinitialiser complètement le poste et de se charger quel que soit le contexte de ce dernier.