

***SGBD (Système de Gestion de Base de Données) SOCRATE –
Projet SOCRATE, Nouvelles spécifications (version 3) – 1972.***

Ce document est disponible dans la collection de l'ACONIT, Association pour un conservatoire de l'informatique et de la télématique, Grenoble, **fiche n°24639-01**

Auteurs : Jean-Raymond Abrial (chef de projet), J.P. Cahen, J.C. Favre, D. Portal, G. Mazaré, Robert Morin.

Document de 650 pages, comportant 9 chapitres.

Edition : Université Scientifique et Médicale de Grenoble : Mathématiques Appliquées - Informatique

Date : Septembre 1972.

Le système logiciel SOCRATE est une « banque de données », constituée de plusieurs « bases de données », sur lesquelles travaillent des utilisateurs.

-La première version de spécification, datée d'Août 1970, a débouché sur un prototype universitaire en décembre 1970, prototype qui sera ensuite industrialisé par ECA-Automation, une SSII travaillant pour le compte de la Sécurité Sociale de Grenoble, puis mis en œuvre à l'Hôpital Universitaire de Grenoble pour le traitement informatique et statistique des dossiers médicaux des patients.

-la seconde version introduit des requêtes d'interrogation et de mise à jour permettant à un utilisateur de communiquer des valeurs entre le système et les périphériques alloués.

-avec la version 3, la présente version datée de 1972, les interrogations et mise à jour sur périphériques consistent en appels des programmes de lecture et écriture associés au périphérique.

Ce logiciel fonctionne en mode « compilé » (cf. Chapitre 6 : Compilateur)

Le présent PDF fournit quelques extraits significatifs de cette spécification version 3 :

- Pages de couverture, d'en-tête, de sommaire avec ses 9 chapitres
- Des extraits des pages qui incluent certains des schémas les plus explicites pour la compréhension du fonctionnement du système SOCRATE de SGBD,
- Chapitre 2 : copie partielle pour la mémoire virtuelle, cette organisation des données étant une particularité de ce type de SGBD

Deux autres fichiers PDF présenteront certains des chapitres les plus significatifs de cette spécification, ceux correspondant à certaines caractéristiques spécifiques de ce logiciel (Il est antérieur aux bases de données « relationnelles » ou SQL, lesquelles n'apparaîtront qu'à partir des années 1976) :

- Chapitre 4 : Espaces virtuels – base de données des bases de données
- Chapitre 9 : Organisation d'une base de données. Une copie partielle de ce chapitre montrant les accès de « fichier inverse par chaîne de bits », l'une des particularités de ce type de SGBD.

Pour utiliser ce document, mentionnez Université Scientifique et Médicale de Grenoble, ainsi que ACONIT n° **24639-01**.

Il est rappelé que la loi sur les droits d'auteur n'autorise que les reproductions partielles de documents à titre individuel et pour des motifs de recherche.

Huntan

UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE

Mathématiques Appliquées - Informatique

PROJET SOCRATE ⁽⁶⁾

NOUVELLES SPECIFICATIONS

(Version 3)

par

J.R. ABRIAL, J.P. CAHEN, J.C. FAVRE

D. PORTAL, G. MAZARE ET R. MORIN

Grenoble, Septembre 1972.

Ce travail a été réalisé dans le cadre des contrats

D. R. M. E. N° 71-34-334-00-480-75-01

C. R. I. N° 70-088

Chapitre 1	-	UTILISATEURS
Chapitre 2	-	SUPERVISEUR DE MULTIPLE ACCES
Chapitre 3	-	MECANISME DE PROTECTION SUR LE FICHER
Chapitre 4	-	ESPACES VIRTUELS - BASE DE DONNEES DES BASES DE DONNEES
Chapitre 5	-	LANGAGES
Chapitre 6	-	COMPILATEUR
Chapitre 7	-	LIAISONS AVEC UN SYSTEME D'EXPLOITATION
Chapitre 8	-	FONCTIONNEMENTS ANORMAUX
Chapitre 9	-	ORGANISATION D'UNE BASE DE DONNEES

-:-:-:-:-:-:-:-:-:-:-:-

CHAPITRE 1

1. UTILISATEURS

1.1. Définition de la notion d'utilisateur

1.1.1. Définition intuitive

Le système SOCRATE est une "banque de données", constituée de plusieurs "bases de données", sur lesquelles travaillent des utilisateurs.

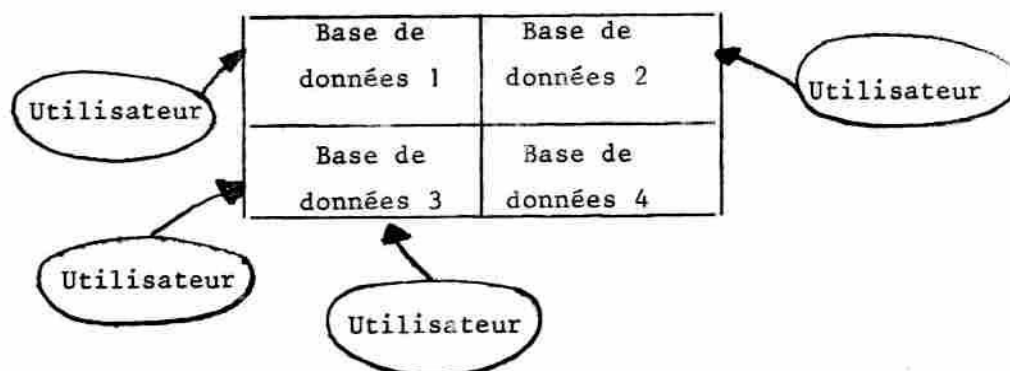


FIGURE 1.1. : Utilisateurs d'une banque de données

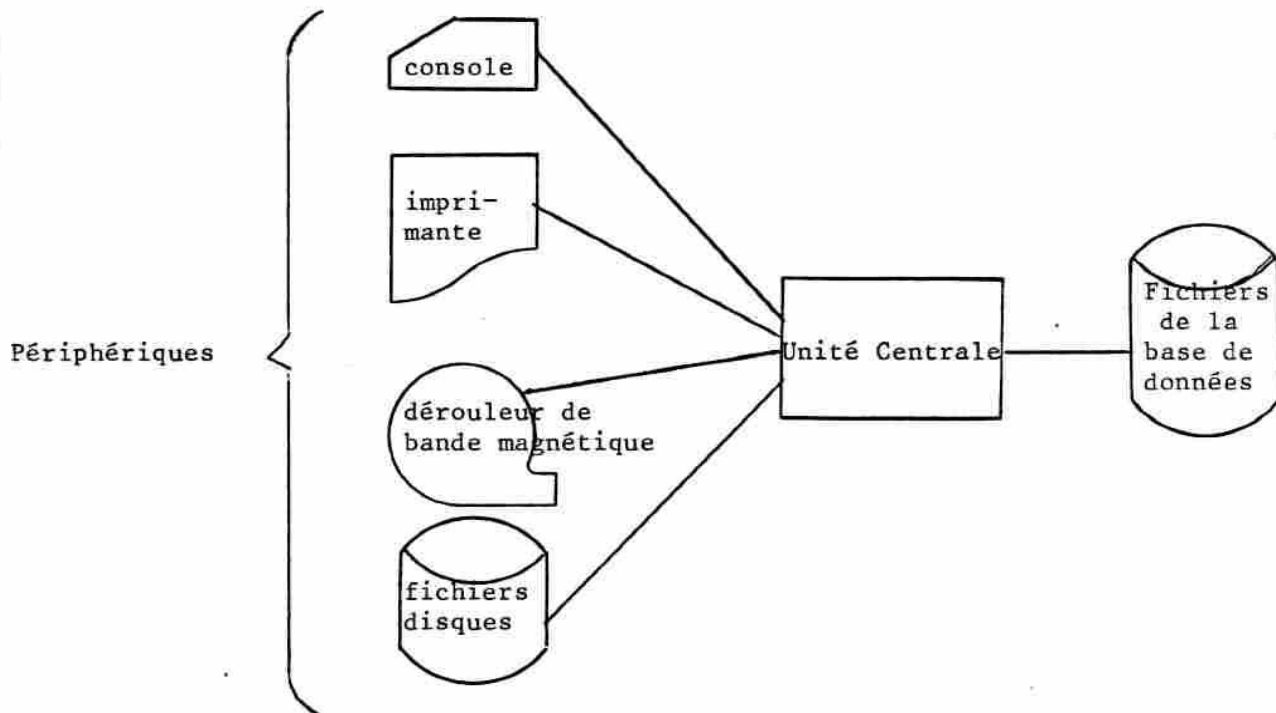


FIGURE 1.2. : Manifestation d'un utilisateur en action

La manifestation externe d'un utilisateur en action est alors l'affectation de périphériques, par le système, à un utilisateur particulier.

1.1.2.2. Utilisateurs potentiels

Une banque de données se compose de plusieurs bases de données. La gestion de ces bases est faite par le système et l'opérateur au moyen d'une base particulière, appelée "base de données des bases de données" (BDB).

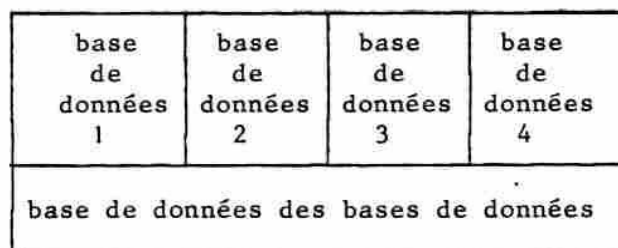


FIGURE 1.3. : BDB

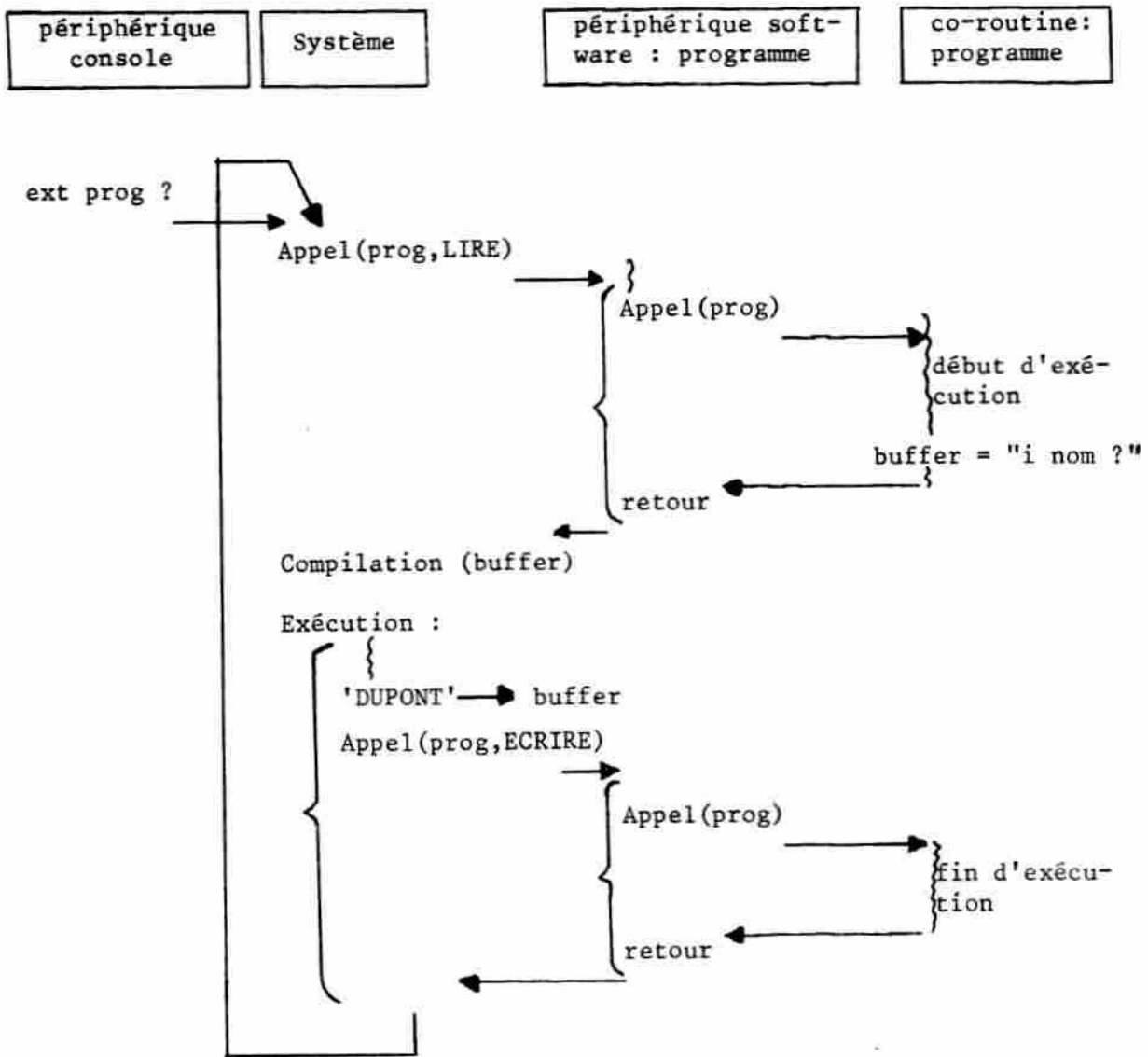


FIGURE 1.14. : Ecriture sur un périphérique logique depuis ce même périphérique logique.

Le périphérique software se distingue donc par la possibilité d'interrompre l'exécution de son programme propre, puis de le reprendre au point d'interruption. Ceci est réalisé par l'utilisation, dans ce programme, de primitives de synchronisation; ceci sera étudié plus en détail au chapitre 7.

2.3.1. Rappels concernant la pagination

2.3.1.1. Mémoire virtuelle

Le principe de la méthode est de fournir aux programmes en aval une mémoire virtuelle très grande, quasi infinie; les programmes pourront donc l'utiliser "statiquement", c'est-à-dire que la place de toute information existante ou non, mais prévue par la structure, y est réservée.

Exemple :

Soit la structure :

```
ENTITE PERSONNE (100)
  DEBUT NOM MOT
    ENTITE VOITURE (5)
      DEBUT MARQUE (PG, CT, SM)
        NUMERO MOT
      FIN
    FIN
  FIN
```

L'espace virtuel sera généralement organisé comme suit :

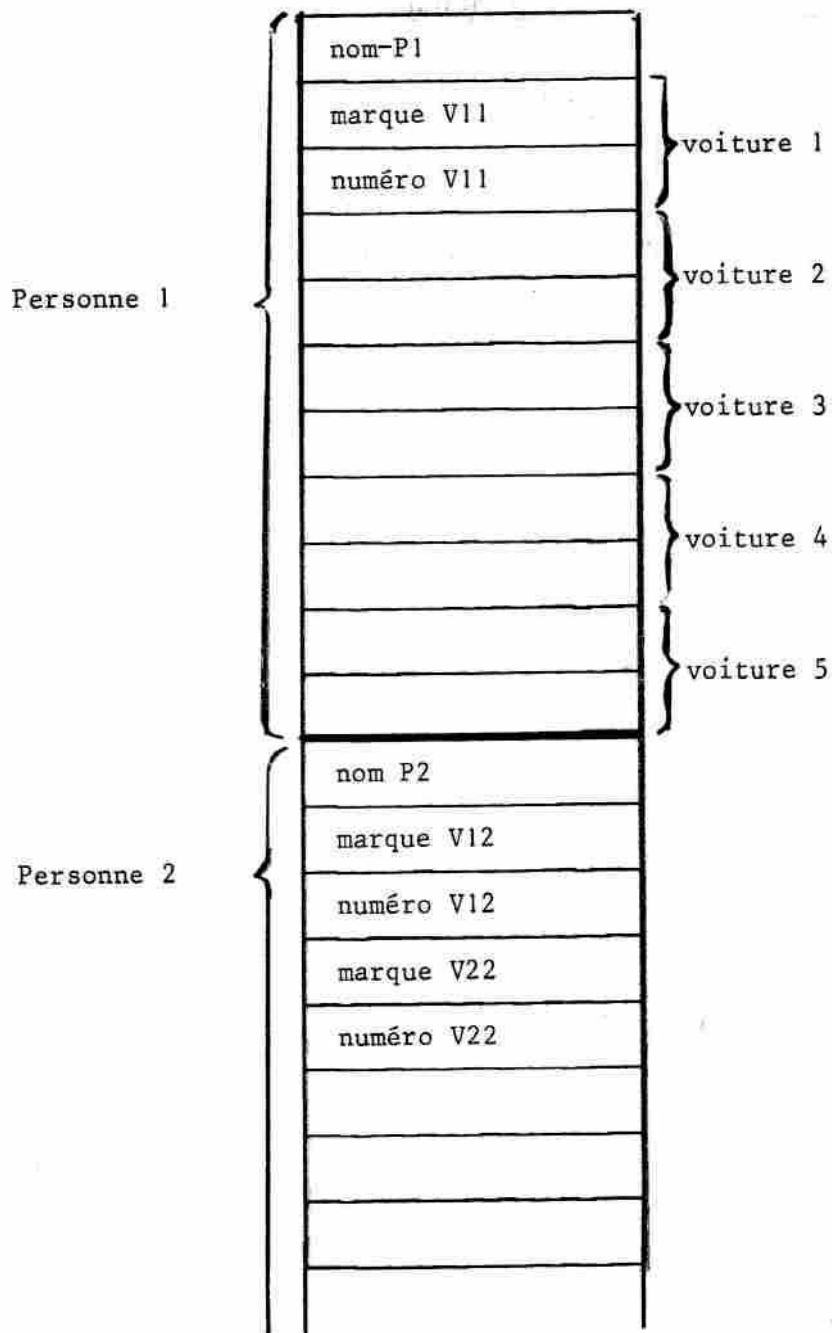


FIGURE 2.16. : L'espace virtuel est organisé de façon statique

Même si la personne P1 n'a qu'une voiture, la place est réservée, en virtuel, pour le nombre maximum de réalisations de l'entité voiture (ici 5).

Les algorithmes de la pagination doivent réaliser la projection d'un tel espace virtuel (immense) sur un espace réel (limité). Cela sera réalisé par les algorithmes :

Ecrire (adresse virtuelle, buffer)
Lire (adresse virtuelle, buffer)

2.3.1.2. Projection de l'espace virtuel sur l'espace réel

La correspondance adresse-virtuelle $\leftrightarrow$ adresse réelle en mémoire secondaire est réalisée par H-code très simple, qui consiste à prendre l'adresse virtuelle modulo la taille de l'espace réel.

Cela veut dire que l'espace virtuel est "replié" sur l'espace réel.

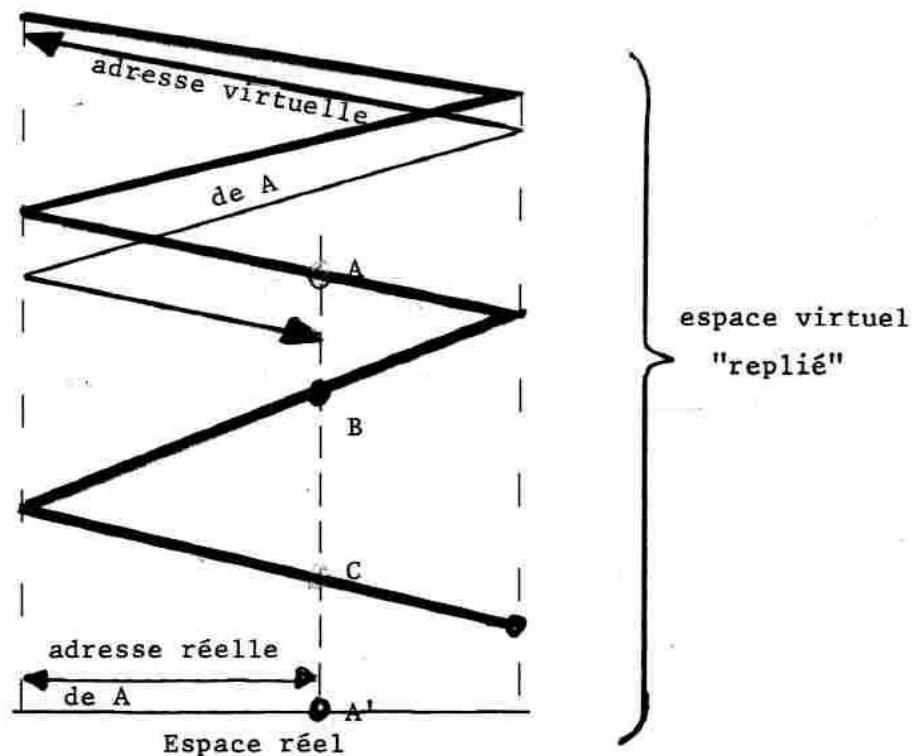


FIGURE 2.17. : L'espace virtuel est projeté par repliement sur l'espace réel

Comme toute méthode projetant un espace grand sur un petit, celle-ci présente des risques de collisions; ainsi les points A, B et C de l'espace virtuel ont tous la même adresse réelle (les restes des divisions des adresses virtuelles de A, B et C par la taille de l'espace réel sont les mêmes).

La méthode de résolution des collisions va consister à employer un dictionnaire, et à chaîner ensemble des points qui font collision. A une entrée dans ce dictionnaire correspond une place dans l'espace réel.

Ainsi, si le hash-code donne pour A, B et C la même adresse réelle, le premier arrivé, par exemple A, sera à sa place; B sera rangé dans une place libre, et un chaînage sera mis en place dans le dictionnaire; etc ...

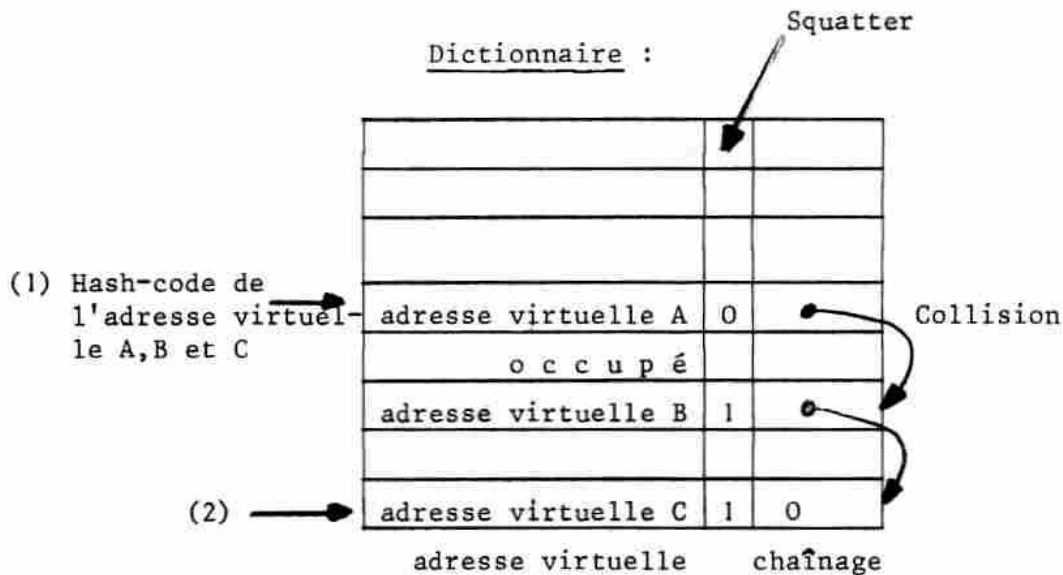


FIGURE 2.18. : Dictionnaire de Hash-code

Quand on recherche B, le hash-code nous donne l'entrée (1) du dictionnaire; la comparaison de l'adresse virtuelle de B à ce que contient la colonne "adresse virtuelle" du dictionnaire n'étant pas satisfaisante, on suit le chaînage, en répétant cette comparaison.

On voit que B et C ne sont pas à leur place dans ce dictionnaire, (2) pouvant être le résultat du hash-code d'une autre adresse virtuelle B et C sont des "squatters" (Cela devra être noté dans le dictionnaire). Et si on veut créer D en (2), il faudra déplacer B, et refaire le chaînage. Ainsi, la tête du chaînage est toujours à sa place. (De même, si l'on efface A, il faut ramener B à sa place et refaire le chaînage).

2.3.1.3. Réalisation du dictionnaire

La projection de l'espace virtuel sur le réel, le hash code, ne sont pas faits pour chaque mot de l'espace virtuel, mais par des blocs comportant n mots.

Un bloc qui est entièrement vide dans l'espace virtuel n'existe pas dans le réel; par contre, dès qu'un bloc comporte un mot, il existe dans le réel. Il faut donc que ces blocs soient petits pour éviter une perte de place dans le réel trop grande.

D'un autre côté le dictionnaire doit évidemment comporter autant d'entrées qu'il y a de blocs dans l'espace réel. Pour des petits blocs, sa taille devient grande, et il ne peut plus tenir en mémoire centrale.

Le dictionnaire se trouve donc dispersé, en mémoire secondaire; puisqu'une entrée correspond à un bloc, on la placera en tête de ce bloc.

Ce bloc s'appelle la sous-page, et l'élément du dictionnaire le mot d'en-tête de sous-page (ce qui fait un mot de perdu par sous-page; d'un autre côté, elles ont intérêt à être petites; il y a un équilibre à trouver).

Ce mot d'en-tête comprend :

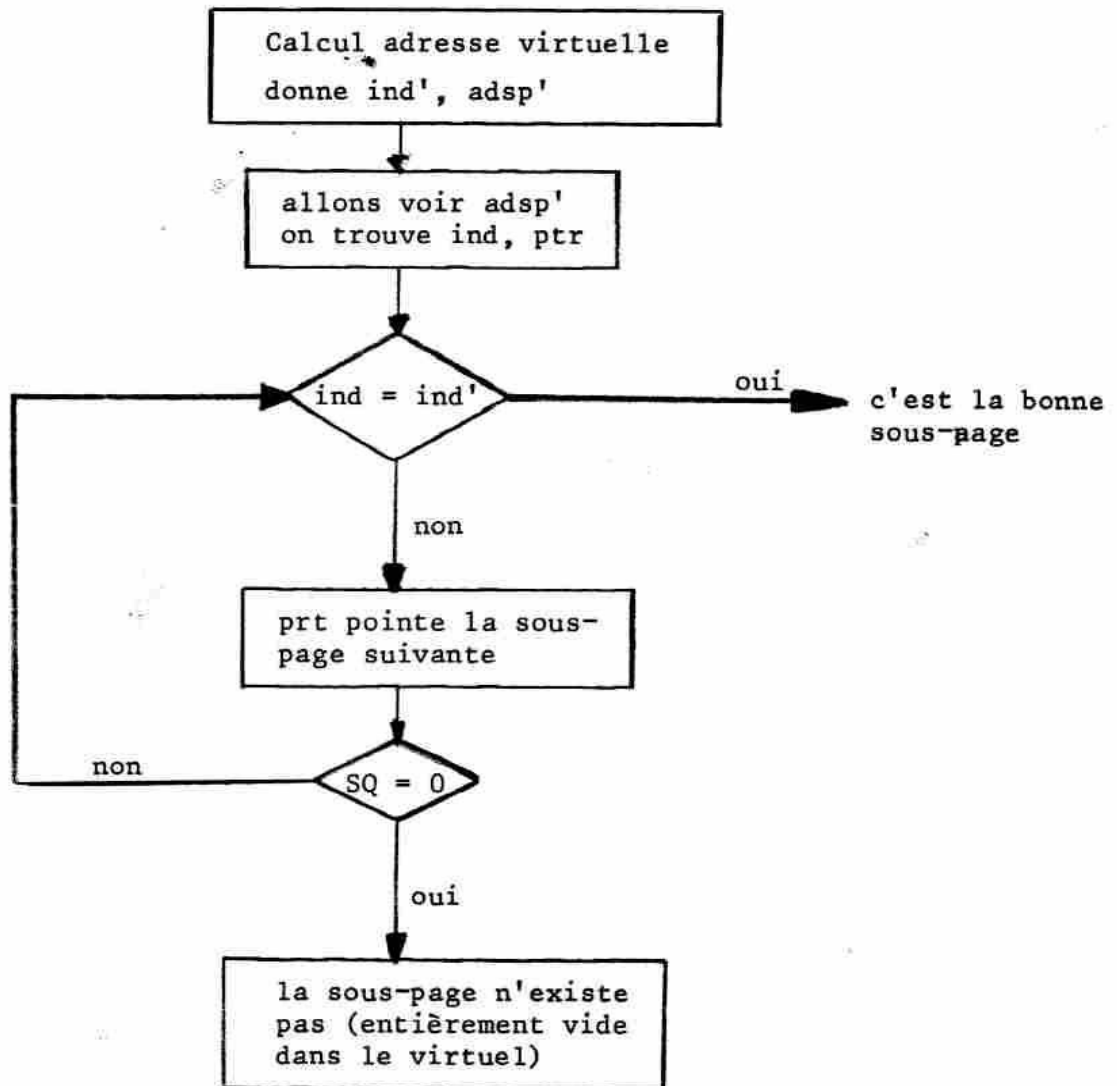
ind	ptr	SQ	N1
-----	-----	----	----

- ind : partie significative de l'adresse virtuelle de départ
(quotient de la division adresse virtuelle/taille espace réel)
- ptr : chaînage vers une autre sous-page
- SQ : bit de "squatter"
- N1 : nombre de lignes occupées dans la sous-page.

L'adresse virtuelle des points A, B ou C n'est pas notée telle quelle, mais devra être recalculée, à partir de ind et de l'adresse de sous-page (le reste de la division). Dans le cas de squatters, leur adresse de sous-page ne veut rien dire (on les a mis là où il y avait de la place); il faut faire le calcul avec l'adresse de sous-page de la tête de chaîne, car celle-ci est toujours à sa place.

2.3.1.4. Fonctionnement de "lire" et "écrire"

- lecture d'une sous-page (adresse virtuelle)
(adsp = adresse de sous-page réelle).



- Ecriture d'une sous-page

Le calcul de l'adresse virtuelle donne ind' , $adsp'$. Allons voir la sous-page réelle en $adsp'$.

- 1 - $SQ = 0$, $ind = ind'$: la sous-page existe et c'est la bonne
- 2 - $SQ = 0$, $ind \neq ind'$: on parcourt la chaîne (nouveaux ind , SQ , ptr)
 - a) - on trouve $SQ = 1$, $ind = ind'$: c'est la bonne sous-page
 - b) - on retrouve $SQ = 0$: la sous-page n'existe pas; il faut en trouver une de libre, l'inclure dans la chaîne et l'écrire.

3 - SQ = 1 : la place est occupée

Il faut trouver une sous-page libre dans l'espace réel, y recopier le squatter, re-arranger la chaîne qui comporte ce squatter. On peut alors écrire dans la place récupérée.

4 - On tombe sur une sous-page libre : on peut écrire

Remarque : Récupération des sous-pages.

Si on écrit autre chose que 0 là où il y avait 0, on incrémente NL; si on écrit 0 là où il y avait quelque chose, on décrémente NL; si NL devient nul, il faut libérer la sous-page.

Pour cela, si d'autres sous-pages sont chaînées derrière, et si elle est tête de chaîne, il faut en recopier une de la chaîne à sa place. (La tête de chaîne doit toujours être au bon endroit). Si elle est squatter, il suffit de réorganiser le chaînage en la sautant.

Puis il faut rendre la sous-page récupérée au pool des sous-pages libres. L'organisation de ce pool est réalisée par pages (cf. § 2.3.2.2.).

2.3.1.5. Conclusion

L'espace réel est donc fractionné en sous-pages; et les demandes d'accès seront des demandes d'accès de sous-pages, par leur adresse (adsp).

Cependant, si les sous-pages sont petites, il n'est pas rentable de faire un appel disque pour rentrer une sous-page. On va plutôt entrer un groupe de sous-pages, consécutives, et les garder le plus longtemps possible en mémoire, en espérant que la sous-page demandée resservira, et que les autres seront aussi demandées.

Cela présente aussi l'avantage, du point de vue système, d'avoir des buffers-disque de taille constante, et commune à tous.

2.3.2. Définitions

2.3.2.1. Le cadre physique

Notre gestion de la mémoire centrale est faite par blocs d'une certaine taille fixe : les cadres physiques.

La mémoire, ressource quasi "continue", devient donc une ressource discrète : nous disposerons, par exemple, de 60 cadres de 1K-octet.

Toutes les allocations de mémoire centrale sont donc faites sur la base de ces cadres physiques.

De la même façon qu'une configuration dispose de N unités centrales ou "processeurs physiques", elle dispose de N' cadres physiques.

2.3.2.2. Les pages

Toutes les E/S disques demandées par la pagination sont faites sur la base d'un groupe de sous-pages : on entre une tranche de l'espace réel, de taille fixe. C'est la page.

La taille de page est celle des cadres physiques; elle doit donc être la même pour toutes les applications (alors que la taille de sous-page pourrait varier).

La page est donc de l'information (à rentrer, ou présente en mémoire), tandis que le cadre physique n'est que son support.

Remarque : D'un autre côté, les algorithmes de la pagination connaissent aussi la page : dans un but d'optimisation, les sous-pages sont cadrées sur des frontières de pages; de plus, l'organisation des sous-pages libres est faite par page (une liste libre par page), et les sous-pages faisant collision seront, dans la mesure du possible, chaînées dans la même page.