

AU COEUR DE *TURBO* PASCAL



PLAN MEMOIRE
FORMAT DES DONNEES
ANALYSE DU CODE
CONTENU DE LA LIBRAIRIE

Editions MNEMODYNE

John COLIBRI

AU COEUR
DE TURBO PASCAL

John COLIBRI

A Marie Odile et sa petite famille

BON DE COMMANDE

MNEMODYNE

26 Rue Lavoisier

75009 PARIS

Tel (14 4) 42 83 18 87

Le manuscrit a été écrit à l'aide de :

MNEMO-TEXT

programme de base

sur IBM PC 192K, carte série

et imprimante LASER WRITER d'Apple, reliée au PC par la liaison série

AU COEUR DE TURBO PASCAL

Le livre a été imprimé par :

IMPRIMERIE CHIRAT

Monsieur Alain CHAUD

Tel (16 1) 42 23 41 12

Ce livre se trouve dans toutes les (bonnes) librairies et boutiques informatiques. Si vous avez des difficultés à vous le procurer, vous pouvez l'acheter chez MNEMODYNE, à Paris, ou le commander en écrivant à :

MNEMODYNE

26 Rue Lavoisier

75009 PARIS

Tel (16 1) 42 83 18 87

John COLIBRI

Ancien élève de l'Ecole Polytechnique

John COLIBRI est élève de l'Ecole Polytechnique et ancien élève de l'Institut PASCAL de l'Institut PASCAL. Vous pouvez lui faire parvenir vos remarques, le contacter ou le remercier 26 rue Lavoisier.

Tous droits de traduction, d'adaptation et de reproduction par tous procédés réservés pour tous pays.

Toute reproduction, même partielle, de cet ouvrage est interdite. Une copie ou reproduction par quelque procédé que ce soit (photocopie, réimpression, bande magnétique, disque ou autre) constitue une contrefaçon passible des peines prévues par la loi du 11 mars 1957 sur la protection des droits d'auteur.

© MNEMODYNE 2011 1987

AU COEUR DE TURBO PASCAL

John COLIBRI

Ancien élève de l'École Polytechnique

Tous droits de traduction, d'adaptation et de reproduction par tous procédés réservés pour tous pays.

Toute reproduction, même partielle, de cet ouvrage est interdite. Une copie ou reproduction, par quelque procédé que ce soit, photographie, microfilm, bande magnétique, disque ou autre, constitue une contrefaçon passible des peines prévues par la loi du 11 mars 1957 sur la protection des droits d'auteur.

© MNEMODYNE SARL 1987

Ce livre est édité par:

MNEMODYNE
26 Rue Lamartine
75009 PARIS
Tel (16 1) 42 85 10 82

Le manuscrit a été écrit à l'aide de:

MNEMO-TEXT
programme de traitement de texte typographique écrit en **TURBO PASCAL**
un **IBM PC**, 192K, carte série
une imprimante **LASER WRITER** d'Apple, reliée au PC par la liaison série

Le livre a été imprimé par:

IMPRIMERIE CHIRAT
Monsieur Alain GIRAUD
Tel (16 1) 42 22 41 12

Ce livre se trouve dans toutes les (bonnes) librairies et boutiques informatiques. Si vous avez des difficultés à vous le procurer, vous pouvez l'acheter chez **MNEMODYNE**, à Paris, ou le commander en écrivant à:

MNEMODYNE
26 Rue Lamartine
75009 PARIS
Tel (16 1) 42 85 10 82

John COLIBRI travaille chez **MNEMODYNE** et anime tous les mois les séminaires **PASCAL** de l'**INSTITUT PASCAL**. Vous pouvez lui faire parvenir vos remarques, le contacter ou le rencontrer 26 rue Lamartine.

1 LIVRES**DECOUVREZ PASCAL SUR APPLE][, //e, //c**

John COLIBRI

Un grand classique pour apprendre le langage PASCAL pas à pas sur Apple. Une mine pour le traitement des fichiers, un exposé complet sur les pointeurs et plus de 30 annexes.

TOPIQUES PASCAL

John COLIBRI

Structure et fonctionnement du PASCAL UCSD, comportant de nombreuses techniques inédites: fichiers dynamiques, codeur LISP, programmation orientée objet, tableaux dynamiques, sauvegarde du tas sur disque, affichage rapide, liaison série. 848 pages, 147 programmes.

RECUEIL PASCALISSIME 1

Recueil des 6 premiers numéros de Pascalissime

RECUEIL PASCALISSIME 2

Recueil des numéros 7 à 12 de Pascalissime

RECUEIL PASCALISSIME 3

Recueil des numéros 13 à 18 de Pascalissime

RECUEIL PASCALISSIME 4

Recueil des numéros 19 à 24 de Pascalissime

BASIC APPLESOFT LEXICUM - Basic, DOS 3.3, PRODOS

Paul MERRY

Manuel de référence alphabétique. Définition, utilisation, exemples pour les instructions et concepts de programmation, édition de texte, traitement de fichiers. 645 pages - 3^{ième} édition

VISICALC TUTORIAL

Christian B DUBOS

Manuel pour apprendre à créer, modifier, imprimer, tester des tableaux, à l'aide du programme VISICALC. Contient 10 exemples de tableaux professionnels commentés. 375 pages.

VISICALC LEXICUM

Christian B DUBOS

Guide de référence alphabétique. Définition, utilisation, exemples pour les commandes, fonctions et concepts VISICALC. 375 pages - 2^{ième} édition.

MAGICALC TUTORIAL

Christian B DUBOS

Manuel pour apprendre à créer, modifier, imprimer, tester des tableaux, à l'aide du programme MAGICALC. Contient 10 exemples de tableaux professionnels commentés. 385 pages.

2 MAGAZINE PASCAL**PASCALISSIME**

Le magazine mensuel PASCAL: programmes, évaluations, bibliographie, initiation, utilitaires. L'index des sujets et un bon d'abonnement figurent à la fin de ce livre.

3 SEMINAIRES PASCAL

John COLIBRI anime tous les mois des séminaires PASCAL. Ces séminaires sont couverts par des conventions de formation. Ils sont basés sur des travaux pratiques en TURBO PASCAL sur IBM PC.

SEMINAIRE D'INITIATION PASCAL

Types de données simples, structures de contrôle, procédures et fonctions, données structurées, gestion de fichiers, tris en mémoire, programmation structurée
4 jours

SEMINAIRE DE PERFECTIONNEMENT PASCAL

Instructions avancées, compilation séparée et segmentation, structures de tris et d'index, saisie contrôlée, gestion de masques, utilitaires PASCAL, versions PASCAL
4 jours

4 PROGRAMMES**MNEMO TEXT**

Traitement de texte français sur Apple][. Multi-fenêtres. Minuscules accentuées sur APPLE][48K standard, avec carte langage. Justification locale, impression proportionnelle ou non, titres. Utilitaires de création d'index et de tables des matières.

MNEMO DUMP

Programme d'examen du contenu des disquettes

MNEMO DE P CODE

Programme de désassemblage du P CODE PASCAL UCSD

5 HOT LINE PASCAL

Support technique PASCAL par téléphone, tous les lundis de 14 heures à 18 heures.
Tel: (16 1) 42.85.10.82

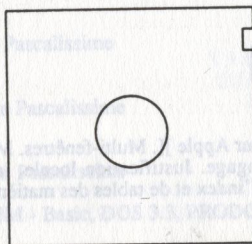
6 CLUB DES UTILISATEURS PASCAL

Réunion hebdomadaire des utilisateurs de PASCAL. Séances d'initiation (exposé et manipulations) et séances de présentation (techniques, matériels, logiciels). Bibliothèque d'échange de logiciels. Tous les mercredis de 19h30 à 21h à l'INSTITUT PASCAL

7 DISQUETTES DE PROGRAMMES SOURCE

Vous pouvez vous procurer les disquettes suivantes, en envoyant un chèque du montant indiqué à MNEMODYNE, 26 Rue Lamartine, 75009 PARIS:

- DECOUVREZ PASCAL 1	75F
- DECOUVREZ PASCAL 2	75F
- PETITS UTILITAIRES PASCAL/APPLE	98F
- TOPIQUES PASCAL	100F
- AU COEUR DE TURBO PASCAL	100F



L'INSTITUT PASCAL a pour vocation la promotion du langage PASCAL.

Il vous propose:

- Plus de 330 livres pour apprendre, utiliser, implémenter PASCAL
- Des revues sur Pascal, parmi lesquelles:
 - PASCALISSIME (les numéros isolés depuis le début, les volumes reliés, les disquettes)
 - The Journal of Pascal and Ada
 - COMPUTER LANGUAGE
 - Dr DOBBS
- Une bibliothèque d'échange de programmes source, comprenant entre autres:
 - le compilateur P4 de Zurich
 - le source de l'interprète de P Code UCSD 6502
 - les compilateurs interprètes Pascal S et Pascal C
 - des utilitaires de désassemblage, dump d'écran ou disquette, de séquentiel indexé, de gestion de menu, de gestion de disquettes...
 - les disquettes de Pascalissime
- Un CLUB D'UTILISATEURS DE PASCAL se réunissant tous les mercredis à partir de 19 heures. Parmi les sujets présentés en 1984 citons:
 - l'Operating System UCSD
 - le Pascal MS sur IBM PC
 - MODULA II
 - le gestionnaire de fichier OMNIS
 - les fichiers de longueur variable en UCSD
 - le séquentiel indexé KSAM
 - DBASE II et la gestion de fichiers
 - démonstration de l'exactitude d'un programme
 - l'éditeur de texte avancé ASE
 - un interprète PROLOG en PASCAL
 - TURBO PASCAL
 - la norme VIDEOTEXT
 - le PASCAL UCSD sur IBM PC sous MS DOS

plus, tous les 15 jours, des séances d'initiation pour débutants, avec manipulations sur micro ordinateur

- des séminaires professionnels de tous niveaux:
 - débutants en informatique
 - approfondissement des techniques Pascal
 - spécialisé sur un sujet (fichiers, implémentation...)

En tant qu'organisme agréé, L'INSTITUT PASCAL est habilité à établir des conventions de formation. Le montant de ces séminaires de 4 jours est déductible des budgets formation permanente.

L'INSTITUT PASCAL
26 Rue Lamartine
75 009 PARIS

Tel (16 1) 42 85 10 82
Metro Cadet ou N. Dame de Lorette
Ouverture: 9-12 13-18 du lundi au vendredi

Le Club des Utilisateurs Pascal a pour objectif de favoriser les échanges entre les utilisateurs Pascal.

Pour cela:

- des réunions ont lieu chaque semaine
- une bibliographie Pascal est mise à jour
- une bibliothèque de programmes est constituée

Les réunions ont lieu chaque mercredi, à partir de 19H. Chaque réunion a un thème, et une présentation formelle d'une heure environ permet de lancer le débat.

Un mercredi sur deux, la présentation est consacrée aux débutants (cours Pascal avec manipulation sur Apple II, IBM PC ou Mac Intosh).

Parmi les sujets déjà présentés ces deux dernières années, citons:

- Le Pascal interprété du Mac Intosh
- Turbo Pascal
- Systèmes experts en PASCAL
- Le langage PROLOG
- Communications en PASCAL
- un interprète LISP écrit en PASCAL
- Séquentiel indexé en PASCAL
- Gestion multifenêtres en PASCAL
- utilisation du BIOS et du DOS du PC en TURBO PASCAL
- utilitaire de conversion UCSD <-> TURBO
- comment utiliser plus de 64K en TURBO

Les thèmes des 4 prochaines semaines sont affichés en permanence à l'Institut Pascal, et seront prochainement annoncés dans Pascalissime.

Après la présentation formelle sont abordés tous sujets intéressant les uns ou les autres, et en particulier:

- les livres publiés
- les articles parus
- les bugs des versions actuelles
- les nouvelles implémentations
- les programmes commerciaux utilisant PASCAL.

Nos membres comprennent aussi bien des étudiants en informatique, des professionnels travaillant en PASCAL pour le graphisme, la gestion ou le calcul scientifique, des professeurs d'université, ou des amateurs ("des gens qui aiment") PASCAL.

La cotisation annuelle est de 290 F TTC et permet d'assister à 52 séances à compter de l'inscription. Vous pouvez assister à une réunion sans vous inscrire, pour vous faire une opinion.

Pour tous renseignements, contactez:

CLUB DES UTILISATEURS PASCAL
26 RUE LAMARTINE
75009 PARIS
Tel: (16 1) 42 85 10 82

L'INSTITUT PASCAL organise tous les mois des stages de formation PASCAL. Au cours des travaux pratiques, chaque stagiaire réalisera de nombreux programmes utilisables professionnellement. Les programmes seront réalisés sur IBM PC ou compatible en utilisant TURBO PASCAL.

1 STAGE D'INITIATION PASCAL

Ce stage est destiné aux débutants. Aucune connaissance de la programmation ou des micro-ordinateurs n'est requise. Le stage présentera l'ensemble des concepts de base de la programmation en PASCAL:

Types de données simples, tests et répétitions, procédures et fonctions, variables globales et locales, paramètres par valeur et par adresse, tableaux enregistrements et ensembles, gestion de fichiers, pointeurs, programmation structurée.

Chaque participant apprendra à:

- écrire et compiler des programmes en PASCAL, comme, par exemple, une gestion de fichier simple avec tri, pilotée par menu
- mettre en oeuvre les techniques de programmation structurée
- développer des logiciels robustes et facilement maintenables

Stages en Janvier, Mars, Mai, Juillet, Septembre Novembre. Durée 4 jours. 4900 F HT

2 STAGE GESTION DE FICHIERS PASCAL

Ce cours a pour but d'acquérir la maîtrise complète d'un ordinateur en PASCAL. Une connaissance élémentaire de PASCAL est nécessaire. Seront présentés:

Organisation mémoire, implémentation des types, formats fichiers, fonctionnement d'un programme PASCAL, gestion des périphériques, traitement des erreurs, gestion des interruptions, appels BIOS et DOS, utilisation du langage machine

Chaque participant apprendra à:

- maîtriser les techniques de gestion mémoire (dépasser 64K), clavier (macros), écran (fenêtres), périphériques parallèle (imprimante) et série (communications)
- mettre en oeuvre des saisies contrôlées par masque, des traitements de fichiers séquentiels indexés
- concevoir et mettre au point des programmes importants

Stages en Février, Avril, Juin, Octobre, Décembre. Durée: 4 jours. 4900 F HT

3 CONDITIONS

Ces stages de formation sont animés par John COLIBRI. Chaque stagiaire emportera le manuel des transparents du cours, la disquette des programmes qu'il a réalisés ainsi que les corrigés des exercices. Un diplôme sera remis à chaque personne. Les dépenses de ces cours sont imputables à la formation professionnelle. Des tarifs de groupe et des conditions spéciales pour particuliers et étudiants sont fournis sur demande. Des séminaires intra-entreprises peuvent être organisés.

Pour recevoir une documentation complète ou pour vous inscrire, contactez:

L'INSTITUT PASCAL - 26 Rue Lamartine 75009 PARIS (Tel: 42.85.10.82)
AU COEUR DE TURBO PASCAL 11 AU COEUR DE TURBO PASCAL

Nous vous commandons (prix port en France compris. Etranger: +20F par produit):

RECUEILS PASCALISSIME

Les recueils regroupant 6 numéros de PASCALISSIME suivants:

- Recueil 1 (numéros 1 à 6), au prix de 120 F TTC
- Recueil 2 (numéros 7 à 12), au prix de 120 F TTC
- Recueil 3 (numéros 13 à 18), au prix de 120 F TTC

ANCIENS NUMEROS PASCALISSIME

Les anciens numéros suivants, au prix de 20 F TTC:

- | | | | |
|------------|-------------|-------------|-------------|
| — numéro 1 | — numéro 7 | — numéro 13 | — numéro 19 |
| — numéro 2 | — numéro 8 | — numéro 14 | — numéro 20 |
| — numéro 3 | — numéro 9 | — numéro 15 | — numéro 21 |
| — numéro 4 | — numéro 10 | — numéro 16 | — numéro 22 |
| — numéro 5 | — numéro 11 | — numéro 17 | — numéro 23 |
| — numéro 6 | — numéro 12 | — numéro 18 | — numéro 24 |

DISQUETTES PASCALISSIME

Les disquettes des numéros suivants, au prix de 40 F TTC:

- | | |
|------------------|------------------|
| — numéro 1 à 3 | — numéro 13 à 15 |
| — numéro 4 à 6 | — numéro 16 à 18 |
| — numéro 7 à 9 | — numéro 19 à 21 |
| — numéro 10 à 12 | — numéro 22 à 24 |

au format: ☐ IBM PC DOS 2.0 au format: ☐ APPLE][UCSD

ABONNEMENT PASCALISSIME

Un abonnement (France Métropolitaine uniquement), à compter du prochain numéro à paraître (tarif du 24 août 1987):

- 12 numéros au prix de 240 F TTC
- 12 numéros et 4 disquettes au prix de 390 F TTC

LIVRES

- Découvrez PASCAL sur APPLE][, //e, //c, Tome 1: 148 F TTC
- Découvrez PASCAL sur APPLE][, //e, //c, Tome 2: 148 F TTC
- La disquette des programmes du Tome 1: 75 F TTC
- La disquette des programmes du Tome 2: 75 F TTC
- La disquette des petits utilitaires: 98 F TTC
- Topiques PASCAL: 290 F TTC
- Les 4 disquettes de TOPIQUES PASCAL: 100 F TTC

Faites parvenir ce bon, accompagné de votre chèque à:

L'INSTITUT PASCAL
26 Rue Lamartine
75009 Paris
Téléphone 42 85 10 82
Métro CADET ou ND de LORETTE

NOM: _____
ADRESSE: _____



L'INSTITUT PASCAL diffuse une gamme de librairies et d'utilitaires qui facilitent la programmation en TURBO PASCAL. Tous ces produits sont en source TURBO PASCAL.

1 LIBRAIRIES TURBO PASCAL

Ces produits contiennent des collections homogènes de procédures TURBO PASCAL permettant l'affichage rapide, le fenêtrage, les appels DOS, la gestion mémoire, l'utilisation des interruptions, et le dépassement des 64K:

- TURBO POWER PLUS (BLAISE): bibliothèque de procédures. 998F TTC
- TURBO PROFESSIONAL (SUNNY HILL): 659F TTC

BLAISE est un peu plus détaillé et pédagogique, TURBO PROFESSIONAL plus compact et rapide.

2 UTILITAIRES TURBO PASCAL

Ces utilitaires facilitent la mise au point de programmes TURBO PASCAL:

- T DEBUG PLUS (TURBO POWER): debugger symbolique permettant d'examiner et changer le nom des variables, de tracer les instructions, de placer des points d'arrêt, d'afficher le texte source pendant l'exécution, d'utiliser l'éditeur TURBO et DEBUG. Disquette, carte de référence, manuel de 80 pages. Nécessite 256K sur PC. 498F TTC
- TURBO EXTENDER (TURBO POWER): permet à un programme d'utiliser les 640K, la compilation séparée, offre plusieurs modèles de larges tableaux, et comprend de nombreux outils (cache disque, analyse d'overlays, encrypteur, générateur). 2 disquettes, manuel de 150 pages. 698F TTC
- TURBO POWER UTILITIES (TURBO POWER): 9 utilitaires: analyse de structure, chronométrage d'exécution, profil, pretty-printer, répéteur de commande, remplacement de chaînes, recherche de modifications, recherche de fichier, super directory. 3 disquettes, carte de référence et manuel. 798F TTC
- TURBO OPTIMISER: supprime le code et les routines de la librairie inutilisés, constitue des librairies d'objet relocatable. 998F TTC
- RUNOFF (BLAISE): formateur de texte paramétrable. 498F TTC

Vous pouvez vous procurer ces produits directement à l'INSTITUT PASCAL, ou en envoyant votre commande accompagnée du montant (plus 20 francs par produit pour les frais d'expédition)

L'INSTITUT PASCAL - 26 Rue Lamartine - 75009 Paris
Téléphone (16 1) 42 85 10 82 - Métro CADET ou ND de LORETTE

— TURBO POWER PLUS (998 F TTC)	— T DEBUG PLUS (498 F TTC)
— TURBO PROFESSIONAL (659 F TTC)	— TURBO EXTENDER (698 F TTC)
	— TURBO POWER UTILITIES (798F TTC)
	— TURBO OPTIMIZER (998F TTC)
	— RUNOFF (498 F TTC)

NOM: _____
ADRESSE: _____

Le PROLOG 1.3 est un système PROLOG complet, permettant l'apprentissage, l'enseignement et l'utilisation du langage PROLOG.

Il vous offre:

- un éditeur de règles pleine page
- une syntaxe simple, lisible, facilitant la compréhension des règles:

```
COUSIN(cousin1, cousin2) ->
  PERE(père1, cousin1)
  PERE(père2, cousin2)
  FRERE(père1, père2);
```

- un mode trace que vous pouvez activer ou supprimer à n'importe quel moment de l'exécution du programme.

Cette trace permet de suivre le déroulement de l'exécution du programme et facilite grandement la compréhension de l'exécution d'un programme PROLOG.

- une explication du raisonnement utilisé pour arriver à une solution.

L'explication permet de comparer l'efficacité de plusieurs solutions, et surtout de vérifier à posteriori les règles utilisées (validation de diagnostics médicaux par exemple)

- des facilités d'entrée sortie:

```
LIS CARACTERE(lettre_suivante)
AFFICHE(Liste_resultats);
```

de calculs arithmétiques:

```
frequence DEVIENT mesure + 3
```

de comparaisons:

```
montant <= 5
nom <> JEAN
```

- la possibilité de rediriger les sorties vers l'imprimante ou un fichier sur disquette

Cette version met en oeuvre le cut, les arbres infinis, le gel.

Le PROLOG 1.3 est actuellement disponible pour l'APPLE 2 (version 64K ou 128K).

Vous pouvez vous le procurer à:

L'INSTITUT PASCAL
26 Rue Lamartine
75009 Paris
Tel 42 83 45 02

LES OBJECTIFS DE CE LIVRE

Le but de ce livre est de décrire le fonctionnement de TURBO PASCAL sur PC. Les principes de base de la programmation sont présentés de manière simple et claire. Les exemples de programmes sont fournis pour faciliter la compréhension des concepts.

CHAPITRE 1

INTRODUCTION

1.1 NOTATION

Les notations utilisées dans ce livre sont les suivantes :

- Les lettres majuscules désignent des constantes ou des variables globales.
- Les lettres minuscules désignent des variables locales.
- Les chiffres désignent des constantes numériques.
- Les caractères entre guillemets désignent des chaînes de caractères.
- Les symboles mathématiques désignent des opérations arithmétiques.

Le langage TURBO PASCAL est un langage de programmation orienté objet. Il permet de créer des programmes qui peuvent être exécutés sur une machine à microprocesseur. Les programmes sont écrits en langage source et sont compilés en langage machine.

Ce chapitre va présenter :

- l'objectif de ce livre
- la structure du livre
- le matériel et les logiciels utilisés

1.2 CONFIGURATION ET LES LOGICIELS UTILISÉS

Pour écrire ce livre, nous avons utilisé :

- un IBM PC, modèle 286
- 480 Ko de RAM
- un clavier AZERTY
- un disque 5.25"
- une carte son
- deux lecteurs de disquette 360K

PC DOS 2.10, avec le système d'exploitation de KEYBER.COM

Les logiciels utilisés sont :

- TURBO PASCAL 2.0
- DOS 2.10
- KEYBER.COM

1.1 LES OBJECTIFS DE CE LIVRE

Le but de ce livre est de décrire le fonctionnement de TURBO PASCAL sur PC.

Cet ouvrage s'adresse aux personnes travaillant en TURBO PASCAL sur PC ou compatible, ayant déjà quelques mois de pratique de PASCAL et de TURBO.

Il décrit en particulier:

- l'organisation mémoire du PC (chapitre 2)
- l'organisation mémoire de TURBO PASCAL (chapitre 3)
- l'implémentation des types de données en TURBO PASCAL (chapitre 4)
- le code généré par le compilateur (chapitre 5)
- les options de compilation (chapitre 6)
- le contenu de la librairie (chapitre 7)
- quelques modifications de TURBO.COM (chapitre 8)

Ces informations sont utiles pour:

- échanger des données entre des fichiers TURBO PASCAL et d'autres programmes (LOTUS, WORDSTAR...)
- optimiser ses programmes PASCAL (sélections d'instructions plus efficaces en place ou en vitesse)
- compléter les explications de la documentation BORLAND
- dépasser certaines limitations actuelles de TURBO PASCAL (barrière des 64K, débordement à la compilation, compilation séparée)
- faire cohabiter en mémoire plusieurs programmes TURBO PASCAL
- modifier certaines routines de la librairie
- comprendre le fonctionnement de certains utilitaires de programmation utilisables avec TURBO PASCAL (TURBO EXTENDER, TURBO POWER+, T-DEBUG, par exemple)
- satisfaire votre curiosité naturelle...

1.2 LA CONFIGURATION ET LES LOGICIELS UTILISES

Pour écrire ce livre, nous avons utilisé:

- un IBM PC, muni de:
 - 484K de RAM
 - un clavier AZERTY
 - un écran monochrome
 - une carte série
 - deux lecteurs de disquette 360K
- PC DOS 2.10, avec chargement automatique de KEYBFR.COM

- TURBO.COM version 3.20A (39731 octets) avec le fichier de messages d'erreur d'origine (1536 octets). Pour la présentation des nombres BCD et des réels 8087, nous avons utilisé les compilateurs BCD et 8087 de la même version.
- le désassembleur 8086 de l'INSTITUT PASCAL pour présenter les listings du code généré par le compilateur. La syntaxe de ce désassembleur est essentiellement celle de DEBUG d'IBM.

Tous les plans mémoire ont été effectués avec cette configuration. Les programmes PASCAL ont été écrits avec la version TURBO mentionnée plus haut. Vous pouvez acquérir le texte de ces programmes sur une disquette disponible à l'INSTITUT PASCAL.

1.3 NOTATIONS

Les programmes présentés comportent:

- en commentaire, à la première ligne, le nom du fichier sur disque contenant le texte du programme. C'est ce nom que vous pouvez utiliser pour charger le texte depuis la disquette de ce livre. Cette disquette est disponible à l'Institut Pascal.
- en gras les mots ayant une signification spéciale en TURBO PASCAL (mots réservés ou prédéfinis)
- des numéros de lignes pour permettre de facilement commenter le texte du programme. Il ne faut, bien sûr, pas les taper si vous éditez le programme.

1.4 AMELIORATIONS

Je tiens à remercier tout particulièrement le Docteur Alain GUERIN qui a bien voulu corriger le manuscrit et proposé des modifications des programmes du chapitre 8.

Les erreurs qui subsistent restent ma responsabilité. Je suis toutefois particulièrement reconnaissant aux lecteurs qui me les signaleraient.

N'hésitez pas à me contacter pour me faire part de vos commentaires, vos appréciations ou vos suggestions. Vous trouverez aussi à la fin du livre un coupon réponse qui vous permettra de m'indiquer les ouvrages, programmes ou utilitaires que vous aimeriez voir publiés. Vous pouvez me joindre par courrier, en téléphonant à l'INSTITUT PASCAL, ou en venant directement 26 Rue Lamartine.

Paris, le 28 Aout 1987
John COLIBRI

CHAPITRE 2

LES DONNEES EN 8088

Ce chapitre va présenter:

- les unités utilisées en informatique
- l'organisation du 8088
- l'organisation mémoire du PC
- les programmes d'examen de la mémoire et des disquettes

A l'issue de ce programme vous serez en mesure:

- d'examiner le contenu de la mémoire
- d'examiner le contenu d'un fichier
- d'interpréter les listings 8088

Le lecteur familier avec le 8088 et avec les programmes d'examen de la mémoire peut sauter directement au chapitre 3 dans lequel nous examinerons l'organisation mémoire de TURBO PASCAL.

2.1 LES UNITES EN INFORMATIQUE

Commençons par définir les unités usuelles en informatique.

2.1.1 LE BIT ET LES REGROUPEMENTS DE BITS

Le bit est l'élément de base en informatique. Il ne peut prendre que deux valeurs, 1 ou 0.

Avec un bit isolé, nous ne pouvons représenter que 2 données:

- par exemple les valeurs numériques 0 et 1
- ou encore les 2 opérations + et -
- ou les valeurs VRAI et FAUX.

Nous pouvons convenir que 0 "représente" +, et 1 "représente" -. Ce qui nous intéresse ici, ce n'est pas la valeur numérique (0 et 1), mais le fait que nous avons un certain nombre d'informations à représenter.

Pour représenter un ensemble de données plus vaste, il faut grouper plusieurs bits. Ainsi, si nous utilisons 2 bits, nous avons 4 cas possibles:

bit 2	bit 1
0	0
0	1
1	0
1	1

et nous pouvons donc représenter 4 données distinctes:

- par exemple les 4 valeurs numériques 0, 1, 2, 3
- ou encore -2, -1, 0 et 1
- ou finalement les 4 opérations +, -, * et /

Et en groupant n bits, nous pouvons représenter 2^n données distinctes.

L'ordre dans lequel sont écrits ces bits a une importance (01 représente une donnée, 10 une autre donnée). Nous numérotions ces bits, et nous les écrirons toujours en plaçant à gauche les bits ayant le numéro (auss appelé le poids) le plus élevé:

Numéro du bit:	3	2	1	0
Valeur du bit:	1	0	1	1

2.1.2 LES BASES DE NUMERATION

L'écriture usuelle des nombres correspond à une décomposition de la valeur numérique en puissances de 10: 2735 correspond à $2*1000 + 7*100 + 3*10 + 5*1$. Et comme 1000 est la troisième puissance de 10 ($1000=10*10*10$), nous pouvons encore dire que 2735 correspond à:

$$2*10^3 + 7*10^2 + 3*10^1 + 5*10^0$$

Nous utilisons donc normalement la "base" 10 (base décimale):

- la base 10 emploie les 10 chiffres 0, 1, 2, 3, 4, 5, 6, 7, 8 et 9
- une valeur numérique s'écrit de façon abrégée en indiquant uniquement les multiplicateurs des puissances décroissantes de 10

De la même façon, nous pouvons utiliser la base 2 (binaire):

- la base 2 emploie les 2 chiffres 0 et 1
- une valeur numérique s'écrit de façon abrégée en indiquant uniquement les multiplicateurs des puissances décroissantes de 2

Ces puissances de 2 sont:

- 2^0 (1),
- 2^1 (2 en décimal),
- 2^2 (4 en décimal),
- 2^3 (8 en décimal),
- 2^4 (16 en décimal),
- 2^5 (32 en décimal) etc

Ecrivons par exemple la valeur décimale 653 en binaire. 653 est égal à:

$$1*512 + 0*256 + 1*32 + 1*8 + 1*1$$

soit:

$$1*2^9 + 0*2^8 + 0*2^7 + 0*2^6 + 1*2^5 + 0*2^4 + 1*2^3 + 0*2^2 + 0*2^1 + 1*2^0$$

La valeur numérique notée 653 en base 10 est donc aussi désignée par

$$1000101001$$

en base 2.

Lorsque nous souhaitons représenter des valeurs numériques en informatique, il est donc commode d'utiliser une représentation dans laquelle la suite de bits correspond à la notation binaire.

Soulignons que cela n'a rien d'obligatoire. En particulier pour représenter des valeurs négatives, il faut bien introduire le signe. Nous pouvons utiliser le bit de numéro le plus élevé, en convenant par exemple que si ce bit a pour valeur 1, la valeur est négative, s'il a pour valeur 0, le nombre est positif. En utilisant par exemple 3 bits:

- 110 représente ainsi -2 (et non pas 6 !)
- 010 représente +2.

En plus des bases 10 et 2, nous utiliserons aussi la base 16 (base hexadécimale):

- la base 16 emploie les 16 chiffres 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E et F
- une valeur numérique s'écrit de façon abrégée en indiquant uniquement les multiplicateurs des puissances décroissantes de 16

Traditionnellement, les valeurs hexadécimales sont précédées du signe \$, ce que nous ferons dans ce livre. Ainsi \$23 correspond à:

$$2 \cdot 16^1 + 3 \cdot 16^0$$

soit 35 en décimal.

2.1.3 LE QUARTET

Nous désignons par quartet (nibble en anglais: "petite bouchée") une suite de 4 bits.

Ainsi 0111 représente un quartet:

Quartet:

Bit numéro: 3210
Valeur du bit: 0111

Un quartet nous permet de représenter 2^4 , soit 16 données distinctes.

Si nous considérons un quartet comme l'écriture binaire d'une valeur numérique, nous pourrions utiliser un nibble pour les valeurs allant de 0 (0000) à 15 (1111).

A l'aide d'un nibble, nous pouvons aussi représenter les chiffres de base du système hexadécimal (base 16). Ainsi le quartet 0111 sera aussi noté \$7 en hexadécimal. En revanche, 1100, dont la représentation décimale est 12, sera noté \$C en base hexadécimale.

2.1.4 L'OCTET

Un octet (ou Byte: "bouchée") est un ensemble de 8 bits.

Par exemple 01011110, que nous écrirons en séparant les bits 4 par 4: 0101 1110 ou encore:

Octet:

Bit numéro: 7654 3210
Valeur du bit: 0101 1110

Un octet nous permet donc de représenter 2^8 soit 256 données différentes.

Si nous considérons que chaque bit de l'octet 0101 1110 correspond à une puissance de 2 (soit 128, 64, 32, 16, 8, 4, 2 et 1), notre octet aura pour valeur en base décimale:

$$0 \cdot 128 + 1 \cdot 64 + 0 \cdot 32 + 1 \cdot 16 + 1 \cdot 8 + 1 \cdot 4 + 1 \cdot 2 + 0 \cdot 1$$

soit 94.

Dans cette représentation, les valeurs décimales extrêmes d'un octet sont donc 0 et 255.

Nous pouvons aussi noter chaque groupe de 4 bits par un chiffre hexadécimal. Nous obtenons la notation hexadécimale d'un octet (dans notre exemple, 0101 1110 se note \$5E).

L'octet est le regroupement normal des bits dans les systèmes dont le micro processeur travaille sur 8 bits de données. C'est le cas du 6502, du 6800 ou du Z80, par exemple. Dans ce type de système chaque case mémoire peut stocker un octet. Dans ce cas la taille de la mémoire se mesure naturellement en nombre d'octets. Et comme les manipulations des bits par 8 sont privilégiées par l'électronique de ces systèmes, l'octet, ou ses multiples, est utilisé comme élément de base pour représenter de nombreux types.

2.1.5 LE MOT

Un mot (ou Word) sera considéré dans ce livre comme un ensemble de 16 bits, soit deux octets.

Un mot permet de représenter 2^{16} données distinctes (65.536 en décimal).

Comme précédemment, nous écrirons un mot en plaçant le bit de numéro le plus élevé à gauche:

Mot

Bit numéro:	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
Valeur du bit:	0	1	0	1	1	1	1	0	1	1	0	0	1	0	0	1

En décimal, ce mot aurait pour valeur:

$$\begin{aligned} &0 \cdot 2^{15} + 1 \cdot 2^{14} + 0 \cdot 2^{13} + 1 \cdot 2^{12} + \\ &1 \cdot 2^{11} + 1 \cdot 2^{10} + 1 \cdot 2^9 + 0 \cdot 2^8 + \\ &1 \cdot 2^7 + 1 \cdot 2^6 + 0 \cdot 2^5 + 0 \cdot 2^4 + \\ &1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 \end{aligned}$$

soit:

$$\begin{aligned} &0 + 16384 + 0 + 4096 + \\ &2048 + 1024 + 512 + 0 + \\ &128 + 64 + 0 + 0 + \\ &8 + 0 + 0 + 1 \end{aligned}$$

ou 24265.

Nous pouvons écrire le mot sous la forme de deux octets. Nous placerons l'octet de numéro le plus faible à droite:

Mot :

Octet numéro: 1 0
Valeur de l'octet: \$5E \$C9 (FAIBLE)

Les microprocesseurs comme le 8086 ou le 68000 effectuent leurs opérations par mots. Cette unité est donc particulièrement importante dans ces processeurs.

2.1.6 LE SECTEUR, LE BLOC, LE CLUSTER

Pour optimiser le temps de transfert entre les disques et la mémoire, les lecteurs stockent l'information par groupes d'octets appelés secteurs. Le secteur est donc la plus petite quantité transférable entre une disquette et la mémoire.

Le nombre d'octets contenus dans un secteur varie d'un constructeur à l'autre. Certains utilisent des secteurs de 128, d'autres des secteurs de 4096... En ce qui concerne l'IBM, il est même possible de choisir un formatage utilisant des secteurs de 256, 512 ou 1024 octets. Par défaut, le PC utilise des secteurs de 512 octets.

Pour réduire la taille des tables qui gèrent la disquette, il arrive que l'Operating System regroupe plusieurs secteurs. Ces regroupements sont appelés des Clusters. Ainsi le MS-DOS utilise des clusters de 2 secteurs.

Finalement le langage que vous utilisez peut gérer ses fichiers en utilisant une autre structure, que nous désignerons par bloc. Turbo Pascal permet la gestion de fichiers par blocs de 128 octets (que Turbo appelle des Records, terme que nous n'emploierons pas).

En résumé:

- le secteur est le regroupement d'octets transférés par le lecteur
- le cluster est le regroupement de secteurs gérés par l'Operating System
- le bloc est le regroupement d'octets gérés par certaines instructions fichier de Turbo Pascal

2.1.7 LE K

Une autre unité souvent utilisée est le K. Contrairement au nibble, à l'octet, au mot ou au secteur qui correspondent à des regroupements effectifs de bits au niveau de l'électronique, le K est uniquement une unité de mesure de capacité (mémoire ou disquette).

Un K correspond à 2^{10} , soit 1024 en décimal.

Ainsi, si un système a une mémoire de 65.536 octets, nous pouvons aussi dire qu'il a une mémoire de 64 K (car 64×1024 est égal à 65.536).

De même une disquette double face de 40 pistes de 9 secteurs de 512 octets a une capacité de $2 \times 40 \times 9 \times 512$, soit $40 \times 9 \times 1024$ ou 360 K.

2.1.8 LE PARAGRAPHE

Le paragraphe est un regroupement de 16 octets.

Par exemple 3 paragraphes contiennent 48 octets.

Cette unité correspond au fonctionnement du 8088 qui utilise pour désigner un emplacement mémoire une adresse exprimée en paragraphes et une adresse exprimée en octets. Ainsi la cellule située à l'adresse 53 peut être désignée comme la cellule au paragraphe 2 et au décalage 21 ($2 \times 16 + 21 = 53$).

2.2 LE FONCTIONNEMENT DU 8088

2.2.1 L'ARCHITECTURE DU PC

Globalement, notre PC comporte:

- un micro processeur 8088 (ou 8086, 80286, 80386...)
- une zone mémoire
- des périphériques (écran, clavier, disquette, et, éventuellement, des imprimantes, des disques durs, des cartes série...)

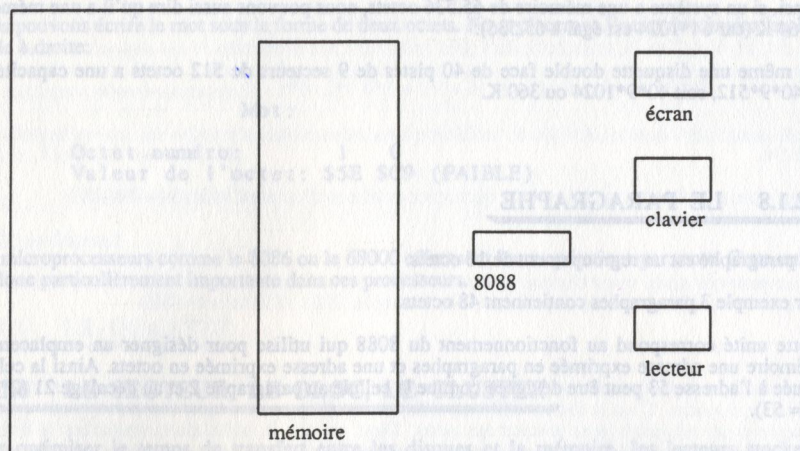


Fig. 2.1 Le micro processeur, la mémoire et les périphériques

Les traitements sont effectués par le micro-processeur, qui communique avec la mémoire et les périphériques par l'intermédiaire de fils, appelés bus.

Le système comporte 3 bus :

- le bus de données
- le bus d'adresse
- le bus de contrôle

Le bus de données est celui sur lequel voyagent les données. Les données transmises sont des octets (8 bits). Le bus de données comporte donc 8 fils.

Le bus d'adresse permet au processeur d'indiquer quelle cellule de la mémoire il souhaite traiter. La capacité mémoire (le nombre de cellules) peut varier entre 64K (65.536) et 640K (655.360), selon les cartes mémoires que nous avons ajoutées. Pour lire ou écrire des données dans l'une des cellules, le 8088 doit préciser l'adresse (le numéro) de la cellule. Cette adresse sera donc un nombre compris entre 0 et 655.360. Pour pouvoir spécifier n'importe quelle valeur dans cet intervalle, il faut utiliser 20 bits (car 2^{20} , soit 1.048.576 est la première puissance de 2 supérieure à 655.360). Le bus d'adresse comporte donc 20 fils. En hexadécimal, une telle valeur utilise 5 chiffres (de \$00000 à \$FFFFF).

Le bus de contrôle sert à préciser la nature de certaines opérations du 8088. Par exemple c'est à l'aide de ce bus que le 8088 indique qu'il souhaite lire ou écrire une cellule de la mémoire.

En nous limitant au 8088 et à la mémoire nous avons donc le schéma suivant :

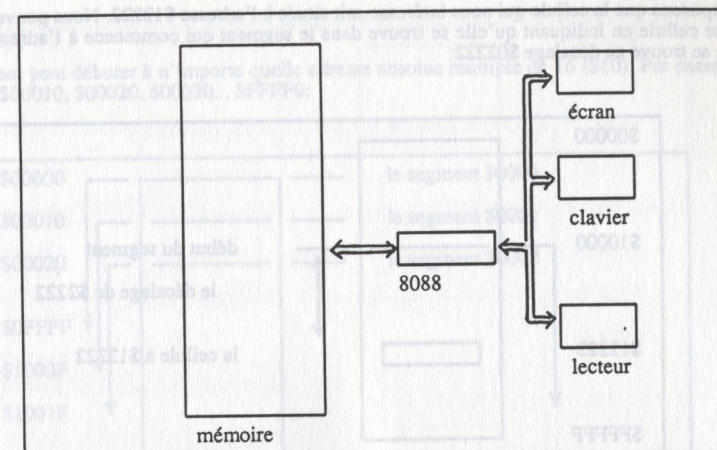


Fig. 2.2 Le micro-processeur, la mémoire et les périphériques

2.2.2 L'ADRESSAGE SEGMENTÉ DU 8088

Fondamentalement le 8088 est capable de traiter des valeurs de 16 bits. Cela lui permet, par exemple, d'additionner deux valeurs de 16 bits pour obtenir un résultat de 16 bits.

Les traitements se font à l'aide des registres du 8088. Ces registres ne sont que des mémoires situées dans le 8088, et dont l'accès est très rapide. Ils sont désignés par AX, BX, CX etc.

Ainsi pour effectuer l'addition de \$1111 et \$2222, le 8088 exécute les opérations suivantes :

```
MOV AX, $1111 ; charge le registre AX avec la valeur $1111
ADD AX, $2222 ; effectue l'addition. AX contient $3333
```

Lorsque le 8088 souhaite accéder à la mémoire, il utilise aussi les registres pour préciser l'adresse visée. Or nous venons de voir plus haut que le bus d'adresse nécessite 20 bits. Il faut donc utiliser au moins deux registres pour spécifier une adresse.

Deux techniques sont possibles :

- l'adressage paginé : un registre précise les 4 bits de poids fort, et un autre précise les 16 bits qui suivent.
- l'adressage segmenté : un registre précise l'adresse de début d'un segment de 64K, et un second registre précise le décalage dans ce segment.

Les concepteurs du 8086 ont choisi la technique de l'adressage segmenté. Examinons donc cette technique plus en détail.

Supposons que la cellule qui nous intéresse soit située à l'adresse \$12222. Nous pouvons accéder à cette cellule en indiquant qu'elle se trouve dans le segment qui commence à l'adresse \$10000 et elle se trouve au décalage \$02222:

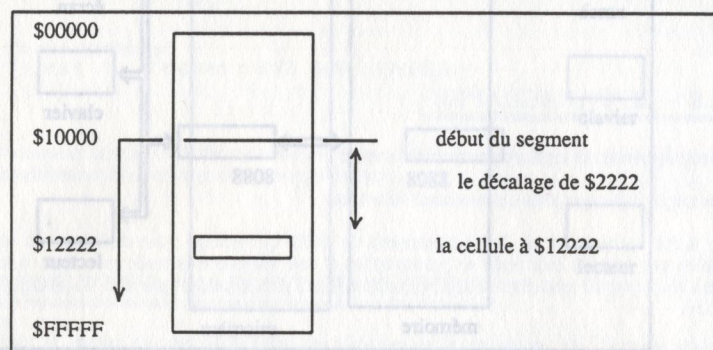


Fig. 2.3 La cellule à l'adresse \$12222

Comme nos registres ne peuvent contenir que 16 bits, Intel a décidé que:

- les adresses de segments ne contiendraient que les 16 bits de poids fort de l'adresse absolue du segment (les 4 bits de poids faible, non explicités, sont par définition 0)
- les décalages ne contiendraient que les 16 bits de poids faible de l'adresse absolue du décalage (les 4 bits de poids fort non explicités sont par définition 0)

Dans notre exemple, le segment aurait pour valeur \$1000 et le décalage \$2222. Traditionnellement, l'adresse est alors notée en indiquant le segment, deux points et le décalage:

\$1000:\$2222

Lorsqu'une adresse segmentée est indiquée, pour obtenir l'adresse complète il faut donc:

- multiplier l'adresse du segment par 16 (\$10)
- ajouter l'adresse du décalage.

Par exemple l'adresse \$3333:\$4444 correspond à l'adresse absolue suivante:

\$33330
\$04444

\$37774

Notez que:

- un segment peut débuter à n'importe quelle adresse absolue multiple de 16 (\$10). Par exemple \$00000, \$00010, \$00020, \$00030... \$FFFF0:

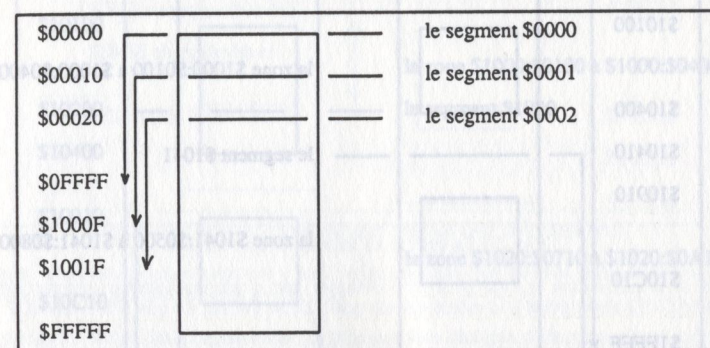


Fig. 2.4 Un segment peut commencer à une adresse multiple de \$10

- l'utilisation d'un segment ne garantit pas qu'il y ait des cellules mémoires pour tous les décalages possibles. Nous pouvons par exemple adresser les cellules du haut de la mémoire par \$FFFF: Décalage, à condition de limiter Décalage à des valeurs entre \$0000 et \$000F. Des valeurs supérieures ne correspondent à aucune cellule mémoire et ne doivent pas être employées

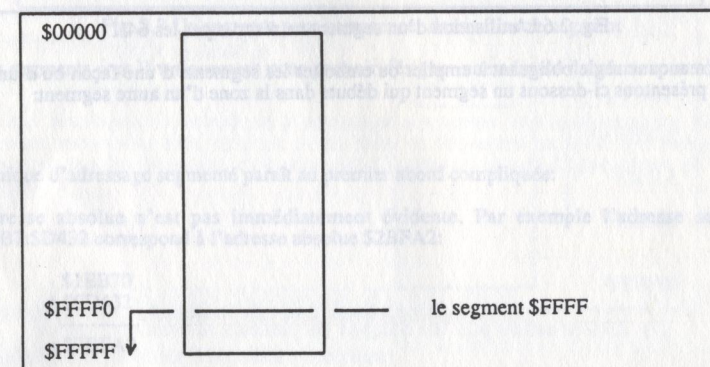


Fig. 2.5 Le dernier segment

- l'utilisation d'un segment n'oblige pas à utiliser tous les 64K du segment. Elle ne "réserve" pas l'espace des 64K: un second segment peut être utilisé pour accéder à des données situées dans la zone du premier. Une partie d'un programme peut très bien utiliser dans le segment \$10000 les décalages \$0100 à \$4000, et une autre partie du programme peut travailler dans le segment \$1410 en employant les adresses \$0500 à \$0800:

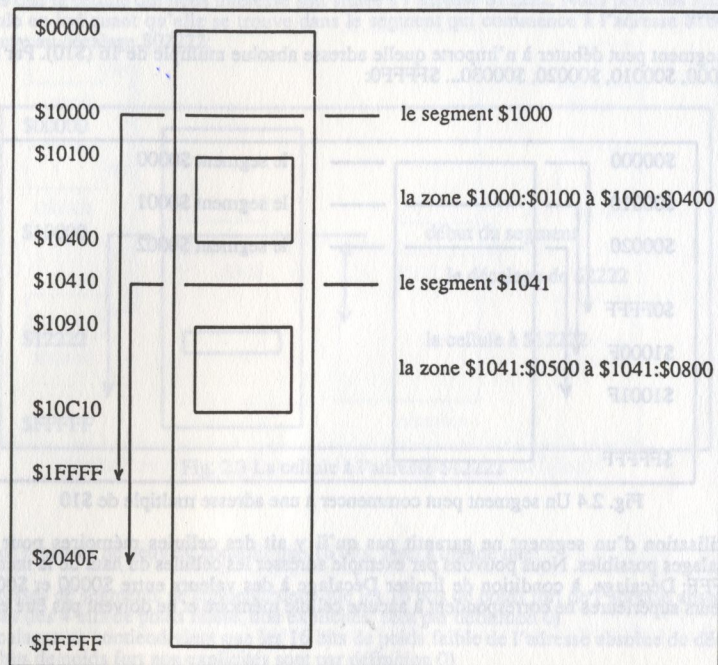


Fig. 2.6 L'utilisation d'un segment ne réserve pas les 64K

- il n'y a aucune règle obligeant à empiler ou emboîter les segments d'une façon ou d'une autre. Nous présentons ci-dessous un segment qui débute dans la zone d'un autre segment:

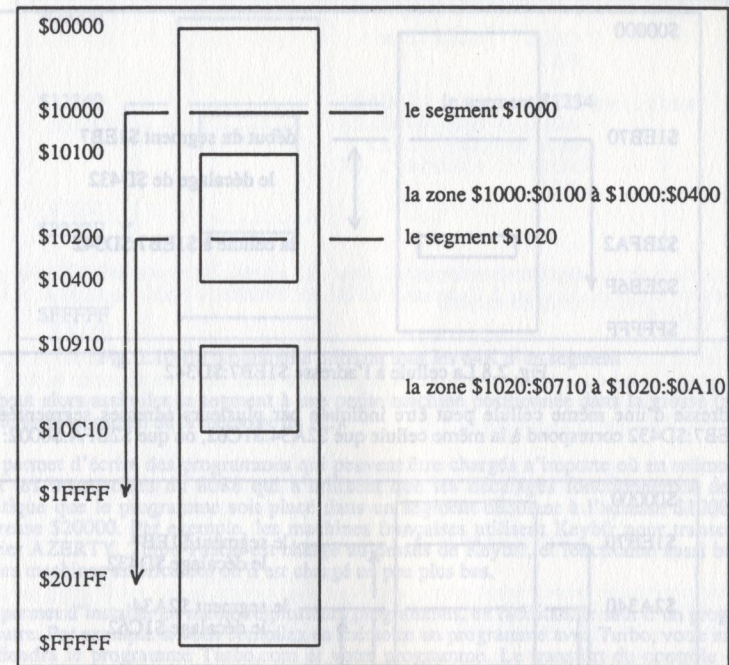


Fig. 2.7 Un segment peut commencer dans un autre segment

Bref, les segments sont là uniquement pour pouvoir désigner une adresse dans un espace de plus de 64K.

La technique d'adressage segmenté paraît au premier abord compliquée:

- l'adresse absolue n'est pas immédiatement évidente. Par exemple l'adresse segmentée \$1EB7:\$D432 correspond à l'adresse absolue \$2BFA2:

$$\begin{aligned} & \$1EB7 \\ & + \$0D432 \\ & \hline & = \$2BFA2 \end{aligned}$$

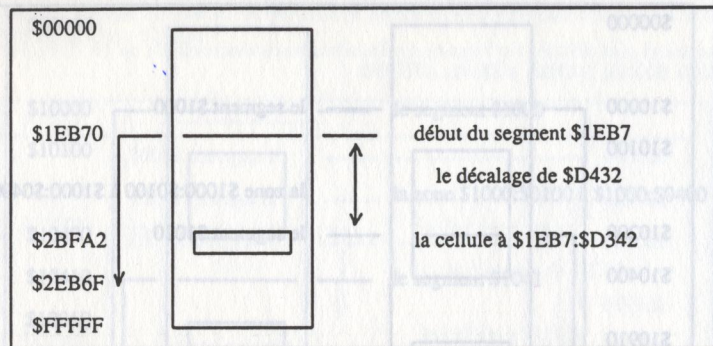


Fig. 2.8 La cellule à l'adresse \$1EB7:\$D342

- l'adresse d'une même cellule peut être indiquée par plusieurs adresses segmentées. Ainsi \$1EB7:\$D432 correspond à la même cellule que \$2A34:\$1C62, ou que \$2BFA:\$0002:

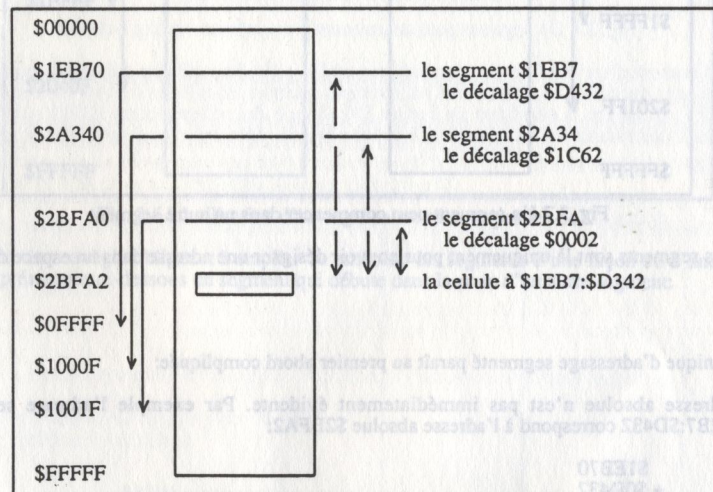


Fig. 2.9 Une cellule peut être désignée par plusieurs adresses segmentées

La technique de l'adressage segmenté a été retenue par Intel pour les raisons suivantes:

- elle facilite l'utilisation des programmes écrits pour les anciens micro processeurs 8 bits du type 8080. Il suffit de sélectionner une zone de 64K de la mémoire, et de ne travailler que dans ce segment:

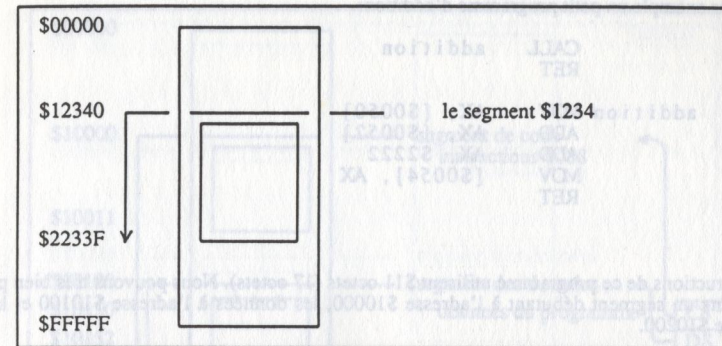


Fig. 2.10 Un programme utilisant tous les 64K d'un segment

On peut alors assimiler le segment à une petite machine positionnée dans la grosse (un petit Apple assis au milieu de la mémoire du PC).

- elle permet d'écrire des programmes qui peuvent être chargés n'importe où en mémoire. En effet les instructions du 8088 qui n'utilisent que les décalages fonctionneront de façon identique que le programme soit placé dans un segment débutant à l'adresse \$10000 ou à l'adresse \$20000. Par exemple, les machines françaises utilisent Keybfr pour transcoder le clavier AZERTY. Turbo Pascal est chargé au-dessus de Keybfr, et fonctionne aussi bien que sur les machines américaines où il est chargé un peu plus bas.
- elle permet d'installer en mémoire plusieurs programmes, en facilitant le saut d'un programme à l'autre. Par exemple si vous compilez en mémoire un programme avec Turbo, votre mémoire contiendra le programme Turbo.com et votre programme. Le transfert du contrôle à votre programme (Run) se fera par un simple changement de segment. Et le retour au menu de Turbo par le transfert inverse.

Malgré ces avantages, la technique d'adressage segmentée explique certaines limitations couramment rencontrées (des tableaux ou du code ne dépassant pas 64K, par exemple). Ces limitations peuvent être tournées, mais nécessitent la mise en oeuvre d'instructions qui changent les segments.

2.2.3 LE CODE, LES DONNEES ET LA PILE

Un programme simple utilise 3 segments principaux:

- le segment de code qui contient les instructions 8088
- le segment de données qui contient les données
- le segment de pile qui permet de gérer facilement les appels de sous-programmes