

BIBLIOTHEQUE DU CERIST

PRINCIPES ET PROBLÈMES D'UN SYSTÈME D'EXPLOITATION D'ORDINATEUR

PAR

J. DU ROSCOÄT

Ingénieur E.N.
Ingénieur à la C^{ie} Honeywell-Bull

MASSON ET C^{ie}, EDITEURS
120, Boulevard Saint-Germain, PARIS (6^e)
— — — — — 1972 — — — — —

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

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective » et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite » (alinéa 1^{er} de l'article 40).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code pénal.

© 1972, Masson et C^{ie}, Paris

Imprimé en Belgique

TABLE DES MATIÈRES

INTRODUCTION	1
CHAPITRE PREMIER. — <i>Fonction d'un système d'exploitation</i>	3
1. — Hardware, firmware, software	3
2. — Les programmes standard	4
3. — Mono- ou multiprogrammation; multitraitement	6
4. — Travail par lots ou en temps réel	8
5. — Travail en temps partagé	9
6. — Structure du système d'exploitation	10
CHAPITRE II. — <i>L'implantation des programmes en mémoire</i>	13
1. — Nature du problème	13
2. — Assemblage de deux sections	14
3. — Section ou segment translatable	17
4. — Sous-programme fermé et translatable	22
5. — Translation du programme entier	24
6. — Sous-programmes ré-entrants	28
7. — Généralisation de la méthode	30
8. — Synthèse des méthodes	32
CHAPITRE III. — <i>Déroulement et interruption des programmes</i>	35
1. — Principe de l'interruption de programme	35
2. — Gestion du travail dans l'ordinateur	36
3. — Causes d'interruption	38
4. — Réalisation de l'interruption	40
5. — Structure du programme de gestion	42
6. — Hiérarchie des interruptions	44
7. — Réparation des erreurs de programme	46
8. — Cas des interruptions normales	48
9. — Exemple de traitement des interruptions normales	49
10. — Notions complémentaires sur les événements	53
11. — Variantes et adaptations	54
CHAPITRE IV. — <i>L'enchaînement des travaux</i>	57
1. — Le but et les outils	57
2. — Début et fin d'un programme	57
3. — Structure d'un programme sur mémoire externe	59
4. — Lecture des segments	61
5. — Rangement d'un programme sur mémoire externe	64
6. — Enchaînement des travaux en monoprogrammation	68
7. — Gestion des travaux en multiprogrammation	70

8. — Exemple concret de « moniteur »	73
9. — Besoins et ressources	74
10. — Choix et décisions par le moniteur	78
 CHAPITRE V. — <i>L'organisation des fichiers.</i>	 81
1. — Le système et les fichiers	81
2. — Fichiers sur supports séquentiels	82
3. — Fichiers sur supports à accès direct	84
4. — Gestion des mémoires à accès direct	86
5. — Organisation d'un fichier sur disques	89
6. — Fichiers séquentiels sur disques	90
7. — Fichiers à adresses calculées	94
8. — Evolution de la notion de fichier	96
9. — Exemple de fichier à plusieurs dimensions	99
10. — Cas particulier des programmes	101
 CHAPITRE VI. — <i>L'affectation et la gestion des appareils périphériques</i>	 105
1. — Fichiers et appareils	105
2. — Les différents niveaux de choix	105
3. — Le dialogue mémoire-appareil périphérique	107
4. — Évaluation des besoins	110
5. — Désignation « manuelle » des appareils	113
6. — Désignation des appareils par le système	116
7. — Organisation rationnelle des fichiers	118
8. — Esquisse à la solution complète	120
 CHAPITRE VII. — <i>La programmation du système</i>	 125
1. — Structure générale	125
2. — Aspect modulaire de l'ensemble	126
3. — Langage de programmation	128
4. — Le PL/1, langage universel	129
5. — Exemple de programmation en PL/1	132
6. — Adaptation du système à la machine	135
7. — Macro-générateur; générateur de système	136
 <i>Index alphabétique</i>	 141

INTRODUCTION

J'AVAIS ÉTÉ FRAPPÉ, il y a quelques années, de voir combien, en pratique, l'enseignement de la programmation à des gens qui comptaient en faire leur métier restait lié à une machine donnée : celle sur laquelle ils étaient appelés à faire leurs premiers travaux. Ensuite, bien entendu, le déroulement de leur carrière les amenait à programmer d'autres machines ; les bons programmeurs remarquaient sans peine les différences et les similitudes et en arrivaient, petit à petit, à dégager la base commune de toute programmation, en la distinguant des détails liés à un matériel particulier.

Certains, néanmoins, restaient assez longtemps imprégnés de leur première formation : ils en éprouvaient une gêne et il leur arrivait parfois d'appliquer plus ou moins consciemment des méthodes un peu étranges, qui ne s'expliquaient que par les caractéristiques particulières d'une machine abandonnée depuis longtemps.

Il m'avait semblé plus logique d'aller du général au particulier, c'est-à-dire d'enseigner d'abord, dans la programmation, ce qui est inhérent au traitement proprement dit des informations, à leur nature, leur structure et même à leur organisation et leurs méthodes de rangement, à condition de faire des réserves sur la permanence de ces méthodes. Ensuite, on pouvait en venir à la façon dont ces traitements étaient réalisés sur chaque machine, grâce à ses organes particuliers (le code d'instructions pouvant être considéré comme un organe parmi d'autres).

De cette idée est sorti l'ouvrage *Conception de la Programmation des ordinateurs*, publié dans cette même collection en 1967. J'y ai exposé le plus possible de méthodes générales ; en tous cas je me suis abstenu rigoureusement de les appliquer à aucune machine déterminée.

La première partie de ce livre (la seconde est plus spécialement consacrée à la description des langages) se termine par un chapitre sur les Programmes standard et systèmes, dans lequel j'essayais de donner une idée de l'importance des programmes que le constructeur livrait traditionnellement en même temps que la machine (), ce qu'on appelle maintenant son software de base. J'ai donné de ces programmes une énumération très incomplète, en indiquant en quelques lignes le rôle de chacun.*

(*) Maintenant l'utilisateur peut en principe acheter séparément la machine et le software, mais cela ne change rien à l'importance de ce dernier.

Ce chapitre m'a toujours laissé très mauvaise conscience, à cause de son caractère succinct. Ses douze pages, dans les 360 de l'ensemble, m'ont paru après coupridicullement courtes pour l'importance du sujet. J'ai donc décidé de le reprendre avec beaucoup plus de détails et c'est l'objet du petit livre que voici.

Ce livre n'est pas la suite du premier, bien qu'à la vérité il le complète. Il en reprend certaines parties en les développant ; de toutes façons, on peut parfaitement le lire isolément.

Mais s'il est moins volumineux que le premier, j'avoue qu'il m'a coûté plus de peine, car le sujet est très difficile à traiter d'une manière générale et indépendante des solutions particulières. Aussi bien ferais-je rire tout le monde en essayant de prétendre que j'ai épuisé le sujet, ou même simplement que j'en ai fait un tour complet. J'ai tenté de classifier un peu les problèmes et de montrer, pour les plus importants, le principe d'une solution, c'est-à-dire de montrer comment pouvaient se traduire, dans la machine, les démarches de l'esprit humain, qui permettent à l'homme d'organiser une activité ; mais je n'ai pu pousser bien loin les exemples d'application de ces méthodes : il aurait fallu descendre à un niveau de détail, que l'on ne pouvait atteindre qu'en faisant des choix très précis, et l'exposé aurait perdu toute généralité. Si le lecteur veut, après ce livre, aborder l'étude d'un système déterminé, il y retrouvera, j'espère, des notions connues et la jungle des séquences qui s'enchaînent et s'appellent mutuellement lui paraîtra peut-être un peu moins inextricable.

Je ne me fais pas non plus d'illusions sur la valeur durable de cette étude. L'évolution de l'informatique est galopante, mais parmi les programmes, le système d'exploitation est encore ce qu'il y a de plus mouvant. On n'en est pas étonné lorsqu'on mesure l'ampleur des problèmes et l'imperfection des solutions. Il est naturel que les créateurs de systèmes soient tenaillés par un perpétuel désir de perfectionner et profitent de toutes les améliorations de la technique de la machine — c'est-à-dire de l'électronique — pour repenser et améliorer leurs produits.

Mais on ne peut pas attendre qu'un outil soit parfait pour le décrire et en enseigner le maniement : si on le faisait, il n'y aurait jamais de livres techniques. Je m'excuse donc du fait que cet ouvrage ne soit que ce qu'il est et je souhaite qu'il puisse néanmoins jeter quelque lumière sur le sujet.