

UNIVERSITE DE GRENOBLE
Faculté des Sciences

LES SYSTEMES D'EXPLOITATION
DES ORDINATEURS



Cours enseigné à l'Institut de Programmation

O.LECARME

1968-69

BIBLIOTHEQUE DU CERIST

TABLE DES MATIERES

Chapitre 1 - INTRODUCTION HISTORIQUE -----	1
1.1 - Introduction -----	1
1.2 - Première generation : 1949-1956 -----	2
1.3 - Deuxième generation : 1957-1964 -----	4
1.4 - Troisième generation : après 1965 -----	6
Chapitre 2 - SYSTEMES A TRAITEMENT SEQUENTIEL -----	9
2.1 - Contrôleur d'entrées et sorties -----	11
2.1.1 - Classification des dispositifs physiques d'entrée et sortie -----	11
2.1.2 - Modes de fonctionnement des dispositifs -----	12
2.1.3 - Moyens de fonctionnement -----	13
2.1.4 - Exemples de sous-programmes d'entrée et sortie	18
2.2 - Assembleur -----	27
2.2.1 - Introduction -----	27
2.2.2 - Caractéristiques principales d'un assembleur	27
2.2.3 - Caractéristiques principales d'un macro-assembleur conditionnel -----	33
2.2.4 - Perfectionnements supplémentaires des assembleurs -----	39
2.3 - Chargeur -----	43
2.3.1 - Introduction -----	43
2.3.2 - Fonctions à assurer -----	43
2.3.3 - Langage fourni au chargeur -----	50
2.3.4 - Chargeur absolu -----	52
2.4 - Compilateurs -----	55
2.4.1 - Passages -----	55
2.4.2 - Phases -----	56
2.4.3 - Exécution du programme compilé -----	57
2.4.4 - Exemples d'organisation de compilateurs -----	58
2.5 - Communication entre programmes -----	61
2.5.1 - Sous-programmes -----	61
2.5.2 - Programmes successifs -----	65

TABLL DES MATIERES

2.6 - Langages de commande -----	67
2.6.1 - Forme des commandes -----	67
2.6.2 - Nature des commandes -----	68
2.6.3 - Traitement des commandes -----	69
2.7 - Mise au point des programmes -----	73
2.7.1 - Analyses de memoire post mortem -----	73
2.7.2 - Interruptions -----	74
2.7.3 - Systemes de mise au point -----	75
2.8 - Generation du systeme -----	81
2.8.1 - Generation sur machine nue -----	81
2.8.2 - Generation a partir d'un autre systeme -----	82
Chapitre 3 - MULTIPROGRAMMATION ET TEMPS PARTAGE -----	87
3.1 - Raisons d'etre de la multiprogrammation et du temps partage -----	87
3.1.1 - Introduction -----	87
3.1.2 - Utilisation de l' unite centrale -----	88
3.1.3 - Utilisation de la memoire -----	89
3.1.4 - Utilisation des peripheriques -----	90
3.1.5 - Utilisation de l'utilisateur -----	91
3.2 - Structure des systemes multiprogrammes -----	93
3.2.1 - Definitions -----	93
3.2.2 - Structures typiques -----	94
3.2.3 - Exemples de systemes -----	103
3.3 - Techniques de multiprogrammation et de partage de temps -----	113
3.3.1 - Dispositifs d'entree et sortie -----	113
3.3.2 - Controle du deroulement des programmes -----	116
3.3.3 - Gestion des ressources du systeme -----	119
3.3.4 - Points divers -----	121
3.4 - Problemes poses par la multiprogrammation et le partage de temps -----	127
3.4.1 - Rentabilite -----	127
3.4.2 - Commodite d'emploi -----	131
Chapitre 4 - TECHNIQUES D'ADRESSAGE -----	137
4.1 - Adressage direct et indirect -----	137
4.1.1 - Adressage simple -----	137
4.1.2 - Adressage complexe -----	137
4.1.3 - Adressage relatif -----	141
4.2 - Translation et registres de base -----	143
4.2.1 - Translation statique -----	143

6

TABLe DES MATIERES

4.2.2 - Translation dynamique -----	145
4.2.3 - Registres de base -----	145
4.2.4 - Adressage symbolique -----	146
4.3 - Pagination et segmentation -----	148
4.3.1 - Pagination -----	148
4.3.2 - Segmentation -----	152
4.3.3 - Calcul d'adresse effective -----	156
Chapitre 5 - CONTROLE DU DEROULEMENT DES PROGRAMMES ---	163
5.1 - Moyens technologiques -----	163
5.1.1 - Etats de la machine -----	163
5.1.2 - Interruptions -----	164
5.1.3 - Horloge -----	167
5.2 - Taches -----	169
5.2.1 - Notion de tache -----	169
5.2.2 - Reentrance -----	171
5.2.3 - Recursiveite -----	171
5.2.4 - Partage des procedures -----	172
5.3 - Gestion des travaux -----	179
5.3.1 - Prise en charge des travaux -----	179
5.3.2 - Repartition des travaux -----	181
5.4 - Comptabilite -----	191
5.4.1 - Facteurs a considerer -----	191
5.4.2 - Methodes -----	195
Chapitre 6 - OPERATIONS D'ENTREE ET SORTIE, FICHIERS --	199
6.1 - Communication avec les peripheriques -----	199
6.1.1 - Peripheriques locaux -----	199
6.1.2 - Terminaux -----	201
6.1.3 - Files d'attente -----	205
6.2 - Methodes d'accès -----	211
6.2.1 - Blocage et deblocage -----	211
6.2.2 - Memoires-tampons -----	213
6.2.3 - Techniques d'accès -----	216
6.3 - Organisation des fichiers -----	225
6.3.1 - Definitions -----	225
6.3.2 - References aux fichiers -----	225
6.3.3 - Securite et protection -----	228

TABLE DES FIGURES

2.1 - Complexite croissante des communications -----	16
2.2 - Sous-programme de lecture de cartes -----	19
2.3 - Sous-programme d'écriture sur bande -----	21
2.4 - Sous-programme d'écriture sur disques -----	23
2.6 - Les chaines de reprise -----	29
2.7 - Recherche par balayage dichotomique -----	31
2.8 - Dictionnaire "hache" -----	31
2.9 - Les divers types de vecteurs de transfert -----	45
2.10 - Structure elementaire -----	47
2.11 - Structure complexe ou une seule chaine peut exister en memoire -----	47
2.12 - Structure complexe ou plusieurs chaines peuvent coexister en memoire -----	48
2.13 - Compilateur mono-passage et mono-phase -----	59
2.14 - Compilateur multi-phases sans entrees ni sorties -----	59
2.15 - Compilateur en deux phases et deux passages ---	59
2.16 - Compilateur en trois phases et trois passages -	60
2.17 - Compilateur en dix phases -----	60
2.18 - Communication des parametres -----	63
3.1 - Systeme sequentiel -----	95
3.2 - Systeme a questions et reponses -----	97
3.3 - Systeme "machine de bureau" -----	99
3.4 - Systeme polyvalent -----	101
3.5 - Systeme conversationnel a un seul langage -----	105
3.6 - Systemes de machines connectees purement sequentiel -----	105
3.7 - Systeme mixte de machines connectees -----	107
3.8 - Systeme polyvalent de taille moyenne -----	109
3.9 - Systeme polyvalent de grande taille -----	111
3.10 - Traitement d'un message -----	122
3.11 - Traitement de plusieurs messages -----	122
3.12 - Construction d'un systeme -----	126
4.1 - Succession de programmes -----	144
4.2 - Pagination a un niveau -----	151
4.3 - Pagination avec table complete -----	153

TABLE DES FIGURES

4.4 - Pagination a deux niveaux -----	153
4.5 - Segmentation sans pagination -----	155
4.6 - Segmentation avec registres (segment courant) --	155
4.7 - Segmentation avec registres (segment exterieur)	155
4.8 - Segmentation avec pagination -----	157
4.9 - Table des segments -----	157
4.10 - Table des pages -----	159
4.11 - Memoires associatives -----	161
4.12 - Calcul d'adresse complet -----	161
5.1 - Priorites -----	166
5.2 - Fonctionnement synchrone et asynchrone -----	170
5.3 - Taches -----	170
5.4 - Structure d'une procedure partageable -----	173
5.5 - Utilisation d'une procedure partageable -----	173
5.6 - La zone de communication -----	175
5.7 - Les mecanismes de communication -----	178
5.8 - Systemes sans priorites -----	184
5.9 - Systeme avec priorites sur une petite machine --	185
5.10 - Systeme simple avec priorites -----	187
5.11 - Systeme polyvalent avec priorites -----	189
6.1 - Chainage des memoires-tampons -----	202
6.2 - Files d'attente en multiprogrammation -----	206
6.3 - File sequentielle -----	209
6.4 - Pile -----	209
6.5 - File a sortie indifferente -----	209
6.6 - Pile a trois priorites -----	210
6.7 - Lecture et ecriture avec mouvement -----	215
6.8 - Lecture avec localisation -----	215
6.9 - Mise a jour par substitutions -----	217
6.10 - Acces par balayage binaire -----	221
6.11 - Acces par table -----	221
6.12 - Effet de la taille des regions sur le nombre de debordements -----	223
6.13 - Effet de l'organisation sur la densite -----	223
6.14 - Repertoire en arbre -----	227
6.15 - Hierarchie des repertoires -----	229

BIBLIOGRAPHIE

Ouvrages generaux.

- J. Arsac
Les systemes de conduite des ordinateurs.
Dunod Universite, 1968.
- J. Bertin, M. Ritout et J. C. Rougier
L'exploitation partagee des calculateurs.
Monographies d'informatique, Dunod, 1967.
- L. Bolliet (editeur)
Traitement a distance en temps reel et en temps partage.
Monographies d'informatique, Dunod, 1967.
- J. Martin
Programming real-time computer systems.
Prentice-Hall, 1965.
- J. Martin
Design of real-time computer systems.
Prentice-Hall, 1967.
- S. Rosen (editeur)
Programming systems and languages.
McGraw-Hill, 1967.

Articles, congres, revues, theses.

- AFIPS conference proceedings.
Voir en particulier le volume 27 (1965).
- ACM national conference proceedings.
Voir en particulier les actes du 22ieme congres (1967).
- IFIP conference proceedings.
Voir les actes du congres de 1968.

BIBLIOGRAPHIE

Communications of the ACM, avril 1968.

Arden, Galler, O'Brien et Westervelt.

Program and addressing structure in a time-sharing environment.
Journal of the ACM, volume 13, pages 1 a 16 (1966).

J. B. Dennis

Segmentation and the design of multiprogrammed computer systems.

Journal of the ACM, volume 12, pages 589 a 602 (1965).

G. Drean

La multiprogrammation - Mythes et realites.

Revue francaise d'informatique et de recherche operationnelle,
numero 9, pages 97 a 116 (1968).

P. J. Denning

Queuing models for file memory operations.

PR MAC-PR-21 (thesis), MIT (1965).

E. L. Glaser et F. J. Corbato

Introduction to time-sharing.

Datamation, volume 10, numero 11, pages 24 a 27 (1964).

F. N. Pyke, Jr.

Time-shared computer systems.

Advances in programming numero 8, Academic press (1967).

A. Rosenberg

Time-sharing : a status report.

Datamation, volume 12, numero 2, pages 66 a 77 (1966).

J. H. Saltzer

Traffic control in a multiplexed computer system.

PR MAC-PR-30 (Ph. D. thesis), MIT (1964).

A. A. Smith

Input/output in time-shared, segmented, multiprocessor systems.

PR MAC-PR-28 (thesis), MIT (1966).

INDEX

acces aleatoire, acces selectif 11, 218, 219, 224
 acces direct 3, 53, 60, 88, 90, 93, 94, 96, 104, 106, 108, 129, 130, 220, 225
 acces sequentiel 53, 210, 218, 222, 224
 adressage 30, 119, chap. 4, 137, 138, 156, 164, 172, 219, 220
 adressage absolu 118, 194
 adressage direct 157, 138, 219
 adressage indice 138, 139, 140
 adressage indirect 44, 137, 138, 139, 140, 142, 147, 158
 adressage relatif 141, 164
 adresse absolue 32, 143, 145
 adresse effective 140, 141, 152, 156, 158, 160
 adresse immediate 143, 141, 156
 adresse réelle 44, 65, 142
 adresse relative 50, 51, 152, 156
 adresse symbolique 28, 146, 150, 152, 154, 156, 158, 172, 174, 176, 177, 200
 adresse translatable 32, 33, 39
 assembleur 5, 27, 30, 32, 33, 34, 35, 36, 37, 38, 39, 40, 43, 44, 46, 51, 57, 58, 65, 68, 76, 77, 78, 81, 84, 102, 103, 141, 143, 218, 230
 asynchrone 12, 13, 169, 171, 204
 bande magnetique 13, 17, 20, 53, 68, 82, 83, 90, 103, 104, 112, 113, 129, 134, 135, 192, 193, 199, 211, 218
 blocage 5, 67, 70, 71, 130, 211, 212, 214, 218
 boucle 34, 39, 138, 186
 canal 12, 13, 15, 17, 18, 20, 22, 24, 82, 83, 115, 116, 121, 129, 162, 169, 186, 191, 201, 203, 205, 208, 218
 catalogue 100, 108, 110, 112, 200
 chaîne 49, 138
 chaîne de reprise 28, 55, 56
 chargeur 30, 32, 34, 39, 43, 44, 46, 49, 50, 51, 52, 53, 58, 65, 68, 70, 77, 78, 81, 102, 143, 145, 172
 codification 17, 110, 219
 commande 24, 67, 68, 69, 70, 71, 100, 120, 200
 communication 56, 62, 64, 65, 70, 88, 102, 106, 114, 172, 174, 176, 182, 192, 199, 201, 212, 230, 231
 compilateur 3, 43, 46, 55, 56, 57, 58, 65, 68, 76, 77, 78, 84, 98, 102, 106, 110, 143, 218, 230
 comptabilité 84, 85, 124, 191, 192, 194, 195, 197
 compteur d'adresses 30, 32, 33, 39, 46
 controleur de ligne 115, 116

conversationnel 93, 94, 98, 103, 106, 108, 123, 200
 dictionnaire 28, 30, 33, 53, 56, 76, 78, 112, 224
 dictionnaire de controle 30, 32, 53, 44, 46, 51
 dictionnaire des symboles 32, 51
 dispositifs d'entree et sortie 6, 8, 11, 15, 17, 58, 65, 70, 82, 85, 88, 104, 110, 123, 129, 167, 188, 191, 193, 195
 disques 13, 22, 66, 82, 88, 90, 94, 96, 112, 113, 120, 130, 149, 169, 192, 195, 199, 200, 205, 208, 211, 218, 222
 editeur 102
 enregistrements 5, 17, 22, 70, 130
 fichiers 66, 67, 68, 70, 88, 98, 100, 106, 108, 110, 112, 113, 121, 130, 132, 133, 134, 135, 182, 199, 200, 201, 203, 204, 211, 212, 214, 216, 218, 219, 220, 222, 224, 225, 226, 230, 231
 files d'attente 90, 96, 98, 106, 121, 130, 167, 181, 182, 183, 186, 188, 194, 203, 205, 207, 208
 fonctions 37, 38
 horloge 115, 124, 163, 164, 167, 183, 192, 196
 interprete 98, 104
 interruptions 13, 14, 15, 18, 20, 24, 74, 75, 78, 104, 106, 108, 116, 117, 118, 121, 133, 145, 147, 180, 182, 162, 164, 165, 177, 188, 192, 195, 205
 langage d'assemblage 24, 34, 36, 39, 57, 67, 75, 76, 81
 langage de commande 51, 104, 110, 112
 langage de programmation 2, 4, 6, 36, 37, 55, 57, 58, 100, 103, 201
 langage source 97, 97, 100, 132
 lien 225, 228, 230
 maillon 40, 52
 memoire 5, 6, 11, 14, 15, 28, 30, 31, 34, 40, 43, 44, 46, 49, 50, 52, 55, 56, 57, 58, 61, 62, 65, 73, 74, 78, 79, 82, 83, 84, 89, 90, 91, 92, 94, 96, 98, 100, 103, 106, 108, 110, 116, 117, 118, 119, 120, 121, 124, 127, 128, 129, 137, 138, 139, 140, 141, 142, 143, 145, 149, 150, 152, 158, 162, 167, 172, 179, 180, 181, 182, 183, 186, 188, 190, 191, 192, 193, 195, 196, 205, 207, 208, 212, 218, 220, 225
 memoire associative 158
 memoire a feuillets 130, 192, 195
 memoire tampon 17, 104, 114, 115, 127, 130, 201, 203, 213, 214, 216, 218
 memoire virtuelle 108, 110, 119, 149, 150, 152, 158
 mise au point 73
 moniteur 106, 108, 110, 113, 129, 172, 188, 225, 226
 multiprogrammation 2, 8, chap. 3, 87, 88, 94, 110, 113, 117, 120, 123, 124, 125, 127, 128, 129, 143, 146, 162, 169, 179, 181, 182, 183, 199, 200, 205, 216, 218
 multitraitement 2, 94, 123
 page 142, 149, 150, 152, 154, 156, 158, 160, 162, 172, 177
 pagination 116, 120, 149, 150, 152, 164
 parametres 18, 20, 22, 24, 36, 37, 38, 40, 61, 62, 64, 65, 67, 69, 70, 71, 75, 110, 112, 172, 174, 176, 177, 182, 196
 pile 38, 61, 74, 117, 141, 172, 200, 207, 208
 passages 30, 55, 56, 58, 118, 132, 164, 194, 204
 peripheriques 5, 84, 90, 91, 92, 94, 113, 116, 120, 121, 127,

INDEX

130, 131, 139, 200, 201, 211, 216, 218, 225
 phase 44, 52, 56, 58, 65, 70, 76, 78, 84, 90, 103, 104, 125, 130, 134, 139
 priorites 14, 33, 93, 98, 100, 104, 121, 128, 129, 130, 143, 165, 182, 183, 186, 188, 190, 191, 193, 194, 196, 204, 208
 procedures 35, 36, 37, 38, 40, 57, 100, 146, 154, 171, 172, 174, 176
 recouvrement 40, 50, 52
 recursif, recursivite 24, 37, 38, 44, 61, 171, 172, 174
 reentrant, reentrance 61, 98, 100, 104, 106, 112, 130, 171, 172
 references symboliques 51, 146, 147
 registres 3, 14, 15, 20, 61, 62, 64, 65, 71, 73, 74, 77, 78, 103, 116, 117, 118, 139, 141, 142, 145, 149, 150, 156, 160, 164, 165, 167, 171, 176, 177
 registre de base 30, 44, 117, 141, 143, 145, 146, 147, 149, 150, 154, 156, 172, 174, 176
 registre d'index 4, 14, 61, 139, 146
 repertoire 82, 133, 139, 200, 201, 225, 226, 228, 230
 section de controle 32, 43, 46, 49, 50, 51, 52, 65, 77
 segmentation 3, 147, 149, 152, 154, 156, 158, 180
 simulation 81, 102, 124, 182, 208
 sous-programme 2, 22, 24, 37, 43, 44, 49, 50, 52, 57, 58, 61, 62, 64, 68, 70, 71, 74, 75, 77, 83, 90, 124, 164, 165, 169, 192, 201, 212, 214
 superviseur 73, 83, 84, 89, 94, 96, 98, 100, 102, 112, 117, 118, 119, 120, 121, 124, 127, 142, 146, 149, 150, 152, 154, 156, 164, 167, 172, 174, 177, 180, 181, 188, 192, 193, 194, 199, 200, 201, 205, 207, 208
 symboles 30, 32, 33, 34, 35, 37, 38, 39, 40, 55, 76, 77, 78, 103, 145, 146, 147
 synchrone 162
 systeme sequentiel 94, 127, 179, 188, 191
 taches 53, 56, 118, 120, 129, 133, 162, 171, 172, 174, 177, 181
 tambour 3, 4, 11, 13, 82, 90, 94, 96, 103, 113, 130, 149, 193, 211, 218
 temps partage, partage de temps 2, 6, 7, 8, chap. 3, 37, 88, 93, 94, 108, 113, 117, 121, 123, 124, 127, 129, 162, 167, 169, 181, 182, 188, 190, 194, 197, 205, 225
 temps reel 2, 15, 94, 121, 123, 169, 181, 183, 188, 205
 terminaux 11, 13, 114, 115, 123, 169, 182, 193, 201, 203
 traitement sequentiel 5, 6, 7, chap. 2, 55, 218
 translation 32, 33, 34, 37, 43, 46, 50, 52, 103, 106, 110, 119, 143, 145, 146, 148, 152, 154, 164, 174, 180, 181
 unite centrale 12, 83, 89, 90, 91, 92, 94, 96, 108, 110, 112, 115, 116, 117, 119, 120, 124, 128, 152, 169, 182, 191, 192, 193, 195, 205
 vecteur de transfert 44, 46

Chapitre 1 - Introduction historique.

1.1 - Introduction.

L'histoire des ordinateurs s'étale pour l'instant sur une période fort brève, puisque, si l'on excepte les tentatives de calcul automatique au moyen d'appareils mécaniques ou électro-mécaniques, cette histoire commence en 1948, avec la construction à l'Université de Cambridge (Massachusetts) de la machine EDSAC. On peut donc encore parler de préhistoire, et penser que les réalisations actuelles qui font notre admiration feront sourire nos enfants, et plus encore ce que nous pouvons tenter d'imaginer de l'avenir des ordinateurs. Cependant l'évolution des matériels et des méthodes d'utilisation a été si rapide et si profonde que l'on peut déjà distinguer trois grandes périodes, que l'on appelle souvent générations quand on parle de la technologie propre des ordinateurs. Chacune de ces périodes se caractérise par un certain nombre de dispositifs technologiques, une certaine vitesse de calcul et un certain mode d'utilisation.

La première génération, de 1949 à 1956, a pour traits marquants le tube électronique ("lampe de radio"), des temps d'opération évalués en millisecondes ou en centaines de microsecondes, et la programmation directe, c'est-à-dire dans le langage même de la machine. Les machines marquantes de cette période sont par exemple les IBM 701 et 650 ou les Univac I et II.

La deuxième génération, de 1957 à 1964, a pour traits marquants le transistor, des temps d'opération évalués en dizaines de microsecondes, ou même en microsecondes, et surtout l'apparition des langages de programmation à haut niveau (Fortran, Algol, Cobol) et des premiers systèmes superviseurs. L'IBM 704 assurant la transition, les machines marquantes sont les séries 7000 d'IBM ou 1100 d'Univac.

La troisième génération a commencé en 1965 et se poursuit actuellement. Elle est marquée par les circuits intégrés, les temps d'opérations en centaines de nanosecondes, et l'inflation extraordinaire de l'importance des systèmes superviseurs, avec les

Chapitre 1 - INTRODUCTION HISTORIQUE

notions nouvelles de multi-programmation, multi-traitement, temps reel, temps partage et traitement a distance.

Nous allons maintenant revoir plus en detail l'histoire de ces vingt dernieres annees pour tenter d'en degager les traits marquants, et peut-etre saisir un peu le sens de cette evolution.

1.2 - Premiere generation : 1949-1956.

Les premiers systemes de programmation furent des systemes de sous-programmes. La machine electro-mecanique Mark I, construite par IBM et l'Universite de Harvard d'apres les idees de Charles Babbage, utilisait deja des sous-programmes. La machine EDSAC construite a l'Universite de Cambridge en 1949 et qui fut le premier calculateur electronique a programme enregistre faisait de la bibliotheque de sous-programmes une des bases de la programmation.

Le calculateur programme par cartes (CPC) d'IBM etait un systeme intermediaire entre les machines comptables mecanographiques et les ordinateurs. Il etait plutot lent mais extremement souple puisque les codes d'instructions etaient determines par des tableaux de cablage modifiables. On pouvait donc soit fabriquer des tableaux qui optimisaient tres bien l'execution d'un probleme precis, soit en fabriquer qui puissent servir a un grand nombre d'applications. Un ensemble de tableaux generaux constituait alors un langage pour le CPC, et c'est ainsi que les utilisateurs scientifiques voyaient generalement la machine comme un calculateur a trois adresses, travaillant en virgule flottante et pourvu d'un ensemble de fonctions telles que racine carree, sinus ou exponentielle. L'experience acquise sur cette machine assez speciale a beaucoup servi pour les systemes de programmation qui ont suivi.

Une autre machine importante dans cette evolution est le Whirlwind (tourbillon) construit au M.I.T. entre 1947 et 1951. Il s'agissait d'une machine rapide mais extremement simplifiee, avec des mots de 16 bits et un repertoire d'instructions tres reduit, qui rendait la programmation en langage machine presque impraticable. Il fallut donc rapidement fournir un moyen de l'utiliser plus commodement ; ce fut d'abord un systeme interpretatif destine aux etudiants, qui aboutit a un systeme general.

Le premier gros calculateur distribue commercialement fut l'Univac I, machine decimale et alphanumerique a mots de douze caracteres. Bien que la programmation directe fut assez facile et que les programmes d'assemblage fussent lents etant donnee la capacite assez faible de la memoire (1000 mots), on fabriqua pour cet ordinateur tout un ensemble de langages de programmation, dont en particulier un langage d'assemblage avec appels de sous-programmes pour les operations en virgule flottante, un langage algebrique (Mathmatic) qui a fourni des idees a Algol, et un langage de gestion (Flowmatic) qui est une des sources les plus importantes de Cobol. Il y eut en plus un programme de tri et un programme important de manipulation de symboles.

Chapitre 1 - INTRODUCTION HISTORIQUE

Un autre groupe d'Univac travaillait sur un langage qui generalisait a l'extreme la notion de procedure, et qui les conduisit a l'etude de notions aussi importantes que la segmentation des programmes, l'allocation de memoire, et le fait de considerer l'ordinateur comme l'ensemble de la partie purement technologique et du systeme de programmation qui permet de l'utiliser.

L'ordinateur IBM 701 fut le premier ordinateur commercial binaire. On defini pour lui un langage d'assemblage (Speedcode) qui faisait paraître cette machine binaire a une adresse travaillant en arithmetique fixe comme une machine decimale a trois adresses, travaillant en virgule flottante et disposant de registres d'index. Un autre systeme, PACT, fut prevu pour le 701 par un groupe d'utilisateurs, et ne fut termine qu'alors que l'ordinateur etait deja perime.

L'IBM 650 etait un ordinateur de taille moyenne centre autour d'un tambour magnetique, comme il en a existe beaucoup vers 1955, et qui eut beaucoup de succes. On lui donna un systeme interpretatif de travail en virgule flottante qui etait la suite logique des systemes de tableaux de connexions du CPC et du langage Speedcode du 701. Un des problemes principaux poses par les machines a tambour etait l'optimisation des programmes par une repartition judicieuse des instructions sur les pistes. Cette optimisation etait tres ennuyeuse a realiser pour le programmeur, mais tres benefique pour le deroulement des programmes, et conduisit a la realisation d'un assembleur symbolique (SOAP) qui se chargeait de cette optimisation. On fabriqua aussi un compilateur algebrique, IT, qui produisait du langage SOAP, et eut une grande influence sur les systemes ulterieurs.

Durant toute cette periode, la methode d'utilisation des ordinateurs n'avait pas varie. Chaque utilisateur venait lui-meme avec son paquet de cartes ou sa bande perforée, introduisait son programme en machine, le faisait executer et repartait avec ses resultats. Une bonne partie de la mise au point du programme se faisait pendant le passage en machine (a condition que l'utilisateur se soit reserve suffisamment de temps), par interaction directe entre le programmeur et l'ordinateur. Les methodes de mise au point etaient orientees vers cette utilisation directe : deroulement du programme instruction par instruction, systeme d'analyse imprimant le contenu des registres importants apres chaque instruction, examen direct du contenu de la memoire au moyen du panneau de controle ou meme d'un oscilloscope. Il n'y avait aucun intermediaire entre la machine et l'utilisateur, celui-ci pouvant interrompre le deroulement du programme des qu'il constatait une anomalie. Une bonne partie du temps de travail etait remplie par le temps de reflexion du programmeur devant la machine arretee ou par des manipulations diverses au pupitre ; en moyenne, la machine elle-meme ne travaillait que pendant la moitie du temps, ce qui donnait apres tout un rendement acceptable tant que l'installation n'etait pas saturee.

Chapitre 1 - INTRODUCTION HISTORIQUE

Les caracteristiques principales de la premiere generation ont donc ete, dans le domaine de la technologie, l'utilisation des tubes a vide et des tambours, dans le domaine de la programmation l'importance des problemes d'optimisation et la definition de langages d'assemblage symbolique, dans le domaine de l'utilisation le travail en communication directe avec la machine.

1.3 - Deuxieme generation : 1957-1964.

L'IBM 701 utilisait une memoire a tubes electrostatiques qui fonctionnait assez mal. L'apparition des memoires a ferrites magnetiques conduisit a la construction de l'ordinateur 704, qui comprenait dans sa technologie un grand nombre de dispositifs qui etaient auparavant simules par les programmes d'assemblage : les plus importants etaient l'arithmetique en virgule flottante et les registres d'index. Pour accompagner ces nouveautes technologiques, la compagnie IBM decida de fournir un nouveau langage de programmation. Sous la direction de John Backus, se fit un travail d'a peu pres 25 hommes-annees, le but etant de produire un compilateur qui accepterait des formules arithmetiques ecrites sous la forme mathematique classique, et qui fabriquerait un programme au moins aussi bon que celui que pouvait ecrire un bon programmeur. On obtint ainsi Fortran, qui fut distribue plus tard que prevu, et commença par tres mal marcher, mais produisait en effet des programmes remarquablement bien optimises, au moins dans la plupart des cas, au prix d'une compilation extremement lente.

Il fallut attendre Fortran II pour que le compilateur fourni soit un peu plus rapide, et des lors le langage connut une extension enorme, jusqu'a s'imposer a tous les autres constructeurs de machines, et a devenir le langage de programmation de tres loin le plus utilise. Cependant cette evolution fut assez lente, et en 1958 on ne se doutait encore de rien. Quand il fut question, d'abord dans l'association europeenne GAMM puis en liaison avec l'ACM, de definir un langage algebrique international, on partit sur des bases entierement nouvelles et avec des objectifs assez ambitieux. Le premier resultat, apres une reunion a Zurich au printemps 1958, fut le rapport preliminaire sur un langage algebrique international, connu a present comme Algol 58. Au meme moment apparaissaient les calculateurs utilisant des transistors, qui rendirent en quelques mois completement perimes tous les ordinateurs utilisant des tubes a vide. Pour chaque nouvelle machine on voulut definir un langage algebrique qu'on appela Algol, et il y eut bientot autant d'Algois que de machines (Bilgol, Algo, Jovial, Neliac, Mad). Pendant ce temps le comite Algol continuait ses travaux, qui aboutirent apres une reunion a Paris en Janvier 1960, a la publication du rapport definissant le langage Algol 60.

Parallelement, a l'instigation du Secretariat de la defense des Etats-Unis, le plus gros des utilisateurs de machines, fut forme le comite CODASYL, afin de definir un langage de gestion. Ce comite

travailla avec des idées preconçues sur l'utilité de la langue anglaise comme langage de programmation, peu d'expérience sur les langages de gestion et leur compilation, et défini hâtivement le langage Cobol, qui devait initialement être un simple avant-projet, et qui fut finalement établi comme le seul langage de gestion grâce au (ou par la faute du) gouvernement américain qui imposa l'existence de compilateurs Cobol sur tous les ordinateurs qu'il achèterait.

Pendant le même temps se développaient de façon intensive les programmes d'assemblage qui se généralisaient et se compliquaient, en particulier par l'apparition des macro-instructions. Il fallut bientôt insérer les assembleurs dans des systèmes moniteurs, afin d'accélérer le passage en machine des travaux. Avec l'apparition de l'ordinateur IBM 709, qui introduisait la simultanéité des opérations d'entrée et sortie, les membres de l'organisation SHARE les utilisateurs de machines IBM se réunirent pour définir un système de programmation assez puissant, finalement appelé SOS. Ce fut un échec pour de nombreuses raisons : trop de personnes différentes avaient eu la responsabilité de le définir, le système était fortement orienté vers le travail en langage machine au moment précis où Fortran prenait de l'importance, et rien n'était prévu pour permettre de faire du Fortran avec SOS.

Sur les machines IBM 7090/94, qui furent les gros calculateurs scientifiques les plus utilisés, le système d'exploitation le plus courant fut le système FMS, qui était très élémentaire et utilisait très mal les possibilités de la machine. On commença à cette époque à se demander si les systèmes de programmation devaient être fournis par le constructeur des machines, ou élaborés par les utilisateurs eux-mêmes afin de pouvoir s'adapter à leurs besoins précis. La querelle reste ouverte, car le fait de concevoir le système comme partie intégrante de l'ordinateur a conduit, à cause du très grand nombre de configurations d'installations possibles, à des systèmes monstrueux et impraticables.

En quelques années, les progrès technologiques avaient multiplié la puissance de calcul des ordinateurs par 50 ou plus. Les programmeurs n'ayant pas de leur côté fait de progrès analogues dans la manipulation des machines ou la mise au point des programmes, les anciennes méthodes de travail conduisaient à des rendements de 2 à 3%. Pour éliminer le plus possible les interventions humaines, on introduisit alors des systèmes moniteurs à traitement séquentiel ou "par fournées" : les travaux à exécuter sont emmagasinés sur un organe d'entrée, le programme superviseur résidant en mémoire de façon permanente les fait passer les uns après les autres jusqu'à ce qu'il n'y en ait plus. Cependant la vitesse des organes périphériques n'avait progressé qu'assez peu (multipliée par 3 ou 4), et l'on construisit des systèmes où un ordinateur périphérique était chargé d'alimenter le calculateur principal à grande vitesse en s'occupant seul des dispositifs lents (lecteurs de cartes et imprimantes). Les systèmes de programmation devaient également gérer les opérations d'entrée et sortie, pour lesquelles l'introduction de la simultanéité et du blocage des enregistrements donnait trop de

travail aux programmeurs.

Des systemes de ce style sont encore les plus utilises, malgre leurs inconvenients : la memoire est mal utilisee, les dispositifs d'entree et sortie egalement, le temps de reponse varie entre deux heures et deux jours, mais l'utilisation est simple et assez directe. Le plus gros defaut finalement est l'absence totale de communication directe entre l'utilisateur et l'ordinateur.

Les caracteristiques principales de la deuxieme generation ont donc ete, dans le domaine de la technologie, les circuits a transistors et les memoires a ferrites, dans le domaine de la programmation, la multiplication des langages de programmation a haut niveau et des compilateurs correspondants, dans le domaine de l'utilisation, l'importance croissante des superviseurs et le travail purement sequentiel.

1.4 - Troisieme generation : apres 1965.

A partir de 1962, et pour tenter de supprimer l'inconvenient majeur des systemes a traitement sequentiel, c'est-a-dire l'absence totale d'interaction entre le programmeur et la machine, on commença a essayer de faire travailler plusieurs utilisateurs en meme temps : le temps pendant lequel un utilisateur ne fait pas travailler l'ordinateur est utilise pour permettre aux autres de travailler. Les systemes experimentaux de travail en temps partage se multiplierent lentement et se compliquerent rapidement, a la compagnie SDC et surtout dans les diverses universites americaines, en particulier au M.I.T. Tous les systemes realises le furent sur des machines de la seconde generation, qui n'avaient a priori pas ete faites pour cela. Ces experiences permirent de definir un certain nombre de points importants a realiser dans la technologie des ordinateurs a venir et d'objectifs primordiaux a realiser dans les systemes de programmation.

Au printemps 1964, la compagnie IBM annonçait sa nouvelle famille d'ordinateurs, la serie 360. Avec cette serie, la compagnie avait de tres ambitieux projets du point de vue de la programmation. Il y eut d'abord l'introduction d'un nouveau langage a tous usages, qui apres s'etre appele Fortran 6, NPL et MPPL finit par etre PL/1. Alors que la compagnie avait depense des centaines de millions de dollars et des annees dans la conception du 360, la conception de PL/1, qui avait de fortes chances de prendre une importance enorme dans le domaine de la programmation, fut confiee a un comite de six personnes qui devaient terminer leur travail le plus vite possible et ne pouvaient y consacrer que peu de leur temps. Etant donnees les buts ambitieux du langage et la quantite de choses que l'on voulait y mettre, le resultat n'est pas aussi mauvais que l'on aurait pu s'y attendre.

L'effort principal de la maison IBM porta sur le systeme de programmation du 360, le systeme OS. La encore les buts etaient tres ambitieux, et le projet fut soumis a de nombreux changements

Chapitre 1 - INTRODUCTION HISTORIQUE

d'orientation et subit de tres nombreux retards, jusqu'a la livraison, debut 1966, d'une version preliminaire dont les performances etaient au moins decevantes. Depuis, le travail sur ce systeme a continue de facon intensive pour tenter d'ameliorer les performances et les versions se sont succedees a la moyenne de huit par an, dans un effort de programmation qui est de loin le plus important de toute l'histoire des ordinateurs.

Cependant, pour les personnes qui croyaient en l'avenir du partage de temps, l'annonce de la serie 360 fut une lourde deception, puisque aucun des dispositifs technologiques qu'ils avaient demandes n'y figurait. Six mois apres, le M.I.T. annoncait qu'il commandait un ordinateur de la compagnie General Electric, qui comporterait les dispositifs voulus. D'autres laboratoires et universites le suivirent dans cette direction, jusqu'a ce que la compagnie IBM reagisse en annoncant, au printemps 1965, le modele 67 de la serie 360, oriente cette fois-ci vers le partage du temps.

Les projets de systemes adequats suivirent ces annonces de machines : le M.I.T., dans le cadre du projet MAC, devait realiser en liaison avec les compagnies General Electric et Bell Telephone le systeme Multics ; la compagnie IBM, cooperant plus ou moins avec l'Universite du Michigan, devait fournir le systeme TSS 360. Les deux systemes rivalisaient d'ambition, de generalite, d'astuces, de commodites et de performances, et devaient etre distribues fin 67. Tous les exemples trouves dans la breve histoire de la programmation inclinaient au pessimisme relativement a l'observance des delais et aux performances initiales. C'est bien ce qui s'est passe, et malgre le tres grand nombre de personnes, souvent de premier plan, travaillant sur chacun de ces projets, ou peut-etre a cause de cela, on a pu voir ces systemes s'enliser veritablement dans la programmation. Les delais de livraison n'ont pas ete respectes, et les performances actuelles sont particulierement decevantes.

En fait, les meilleurs resultats ont ete obtenus par des projets d'envergure bien moindre : soit des systemes prevus pour de petits ordinateurs, donc tres simplifies et ne desirant pas servir des centaines d'utilisateurs, soit des systemes bricoles rapidement pour repondre aux besoins precis d'un laboratoire, soit des systemes remplaçant la complexite de la programmation par une enorme puissance de travail et de transmission sur l'ordinateur.

Après l'engouement general cause par les premieres realisations experimentales de systemes en temps partage, on s'oriente maintenant vers une vision plus realiste des choses, qui considere que le partage du temps n'est pas une panacee et que les systemes a traitement sequentiel ont encore leur mot a dire, meme sur les ensembles les plus puissants. D'autre part, les ordinateurs de petite taille, que l'on aurait pu croire condammes par l'evolution a plus ou moins breve echeance, manifestent au contraire un regain de vitalite : ils apportent en effet toutes les possibilites de communication directe entre l'homme et la machine, que les gros systemes en temps partage ne semblent pas encore assez murs pour fournir.

Chapitre 1 - INTRODUCTION HISTORIQUE

Pour établir un parallèle avec les deux générations précédentes, les caractéristiques principales de la troisième génération, dont on ne sait pas encore combien de temps elle durera, sont actuellement les suivantes : dans le domaine de la technologie, les circuits intégrés et les nouvelles techniques de mémoires rapides (films minces, fils), dans le domaine de la programmation l'échec sensible des systèmes trop volumineux et la poursuite de la multiplication des langages, dans le domaine de l'utilisation les tentatives de travail en temps partagé et multiprogrammation, le traitement à distance et la grande importance prise par les dispositifs d'entrée et sortie à accès direct.