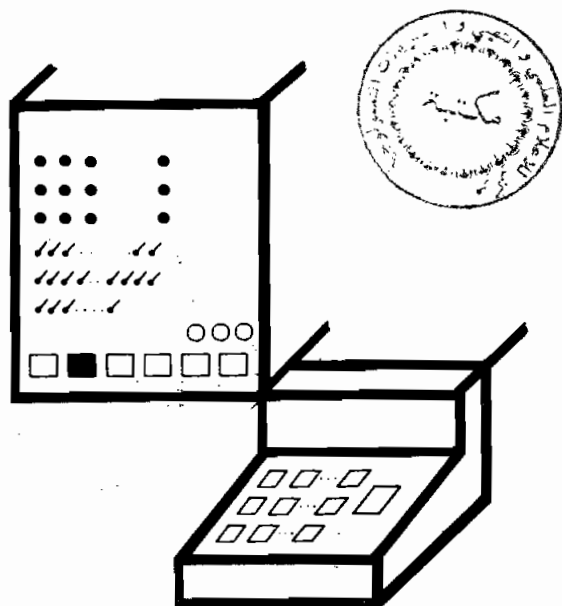


UNIVERSITE SCIENTIFIQUE ET MEDICALE DE GRENOBLE
CENTRE INTERUNIVERSITAIRE DE CALCUL (C.I.C.G.)



*SYSTEMES
DE
PROGRAMMATION GENERATEURS
DE
MACHINES VIRTUELLES*

par

Maurice	BELLOT	
Jean	GUILLOU	
Jean-Pierre	Le HEIGET	C.I.C.G.
Maurice	REY	
Liliane	SIRET	
Jean-Pierre	DUPUY	C.S. IBM

157 293
DECEMBRE 1973

TABLE DES MATIERES

C H A P I T R E I

I	CONFIGURATION REELLE	PAGE 013
I-1	GENERALITES	PAGE 013
I-2	DESCRIPTION DE LA CONFIGURATION	PAGE 015
I-3	LA CONFIGURATION MINIMALE	PAGE 020
II	CONVENTIONS DE LIAISON	PAGE 023
II-1	REGISTRES ET ZONES DE SAUVEGARDE	PAGE 024
II-2	TRAITEMENT DES INTERRUPTIONS SVC. LES MECANISME AUTOMATIQUE DE LIAISON	PAGE 026
II-3	LES MACROS-INSTRUCTIONS (CALL, ENTER, EXIT, GOTO)	PAGE 029
III	MISE EN MARCHE ET ARRET DU SYSTEME CP	PAGE 031
III-1	ECRITURE DU NOYAU ET IPL	PAGE 031
III-2	INITIALISATION DU SYSTEME: LE MODULE CPINIT	PAGE 034
III-3	ARRET DU SYSTEME	PAGE 037
IV	GESTION DE L'ESPACE DISQUE	PAGE 039
IV-1	ATTRIBUTION DE CYLINDRES: LE MODULE TMPSPACE	PAGE 039
IV-2	RESTITUTION DE CYLINDRES: LE MODULE TMPRET	PAGE 040
V	GESTION DE LA MEMOIRE	PAGE 041
V-1	GESTION DES BLOCS DE TAILLE NON STANDARD	PAGE 043
V-2	GESTION DES BLOCS DE TAILLE STANDARD	PAGE 044
V-3	ETUDE DES CAS PARTICULIERS	PAGE 046

C H A P I T R E II

I	DESCRIPTION DE LA MACHINE VIRTUELLE	PAGE 058
I-1	LE CATALOGUE DES MACHINES VIRTUELLES	PAGE 058
I-2	REPRESENTATION DE LA CONFIGURATION VIRTUELLE EN MEMOIRE	PAGE 061

I-3	L'OPERATION DE LOGIN	PAGE 064
I-4	MODIFICATION DYNAMIQUE DE LA CONFIGURATION VIRTUELLE	PAGE 068
I-5	L'OPERATION DE LOGOUT	PAGE 077
II	MISE EN OEUVRE DE LA MEMOIRE VIRTUELLE PAR CP	PAGE 078
II-1	ETUDES DES FONCTIONS NECESSAIRES A LA GESTION DE LA MEMOIRE VIRTUELLE	PAGE 078
II-2	TRAITEMENT DES INTERRUPTIONS-PROGRAMME SPECIFIQUES DE L'UTILISATION DES MEMOIRES VIRTUELLES	PAGE 080
II-3	TRAITEMENT D'UNE FAUTE DE PAGE	PAGE 082
II-4	GESTION DE LA MEMOIRE SECONDAIRE - MODULES PAGEGET ET PAGEREL	PAGE 088
III	L'UNITE CENTRALE VIRTUELLE	PAGE 091
III-1	MODES DE FONCTIONNEMENT DU CALCULATEUR IBM 360	PAGE 091
III-2	MODES DE FONCTIONNEMENT D'UNE MACHINE VIRTUELLE	PAGE 092
III-3	ACTIVATION DESACTIVATION DE LA MACHINE VIRTUELLE - REFLEXION D'UNE INTERRUPTION	PAGE 093
III-4	SIMULATION DES INSTRUCTIONS PRIVILEGIEES	PAGE 098
III-5	TRAITEMENT DES INSTRUCTIONS SVC	PAGE 100
III-6	TRAITEMENT DES INSTRUCTIONS NON PRIVILEGIEES PROVOQUANT UNE INTERRUPTION PROGRAMME	PAGE 101
III-7	RECAPITULATION	PAGE 101
IV	GESTION DES ENTREES-SORTIES	PAGE 102
IV-1	MECANISME DE GESTION DES ENTREES-SORTIES REELLES	PAGE 102
IV-2	RECONNAISSANCE D'UNE DEMANDE D'ENTREE-SORTIE	PAGE 109
IV-3	TRAITEMENT DES INSTRUCTIONS D'ENTREE-SORTIE SUR UN CANAL SELECTEUR VIRTUEL. LE MODULE VIOEXEC	PAGE 110
IV-4	TRAITEMENT DE L'INSTRUCTION SIO SUR UN CANAL SELECTEUR VIRTUEL	PAGE 114
IV-5	TRAITEMENT DES INSTRUCTIONS D'ENTREE-SORTIE SUR UN CANAL MULTIPLEXEUR VIRTUEL. LE MODULE MVIOEXEC	PAGE 126
IV-6	TRAITEMENT DES INSTRUCTIONS SIO SUR LES UNITES DITES DE SPOOLING	PAGE 127
IV-7	TRAITEMENT DE L'INSTRUCTION SIO POUR LA CONSOLE MAITRESSE DE LA MACHINE VIRTUELLE	PAGE 140
V	SIMULATION DU PUPITRE. LES FONCTIONS CONSOLE	PAGE 149
V-1	LA FONCTION STOP	PAGE 150

V-2 FONCTIONS QUI N'AUTORISENT PAS UNE REPRISE D'ACTIVITE DE LA MACHINE VIRTUELLE	PAGE 151
V-3 FONCTIONS QUI AUTORISENT UNE REPRISE D'ACTIVITE DE LA MACHINE VIRTUELLE	PAGE 155
V-4 LA TOUCHE REQUEST DE LA CONSOLE MAITRESSE	PAGE 160
VI COMMUNICATION ENTRE UNE MACHINE VIRTUELLE ET CP. L'INSTRUCTION DIAGNOSE	PAGE 161
VI-1 TRAITEMENT DES INSTRUCTIONS DIAGNOSE. LE MODULE PRIVLGED	PAGE 162
VI-2 LES DIFFERENTES FONCTIONS	PAGE 162

C H A P I T R E III

I INTRODUCTION	PAGE 170
II SYNCHRONISATION DES CHAINES DYNAMIQUES DE TRAITEMENT	PAGE 171
II-1 GESTION DES CPEXBLOK ET REPRISE DE L'EXECUTION DES CHAINES DE TRAITEMENT	PAGE 172
II-2 EXEMPLE D'UTILISATION	PAGE 173
III GESTION DE L'UNITE CENTRALE: LE MODULE DISPATCH	PAGE 175
III-1 ROLE DU MODULE DISPATCH ET DEFINITIONS COMPLEMENTAIRES	PAGE 176
III-2 ETUDE DE CAS PARTICULIERS	PAGE 179
III-3 ETUDE DES SOUS PROGRAMMES FONDAMENTAUX DE DISPATCH ET CLASSIFICATION DES MACHINES VIRTUELLES	PAGE 181
III-4 RESUME DE FONCTIONNEMENT GENERAL DE DISPATCH	PAGE 192
IV GESTION DES MEMOIRES VIRTUELLES. LA PAGINATION	PAGE 195
IV-1 PROTECTION ENTRE MEMOIRES VIRTUELLES	PAGE 195
IV-2 RECHERCHE D'UNE PAGE DISPONIBLE EN MEMOIRE REELLE	PAGE 198
V PARTAGE DE SYSTEME ET PARTAGE DE PAGES	PAGE 202
V-1 IPL PAR NOM	PAGE 202
V-2 UTILISATION PARTICULIERE DE L'IPL PAR NOM: PARTAGE DE PAGES EN MEMOIRE REELLE	PAGE 205
CONCLUSION	PAGE 207

Nous tenons à remercier Madame CHOMAT pour le travail important qu'elle a fourni aussi bien pour la frappe initiale de ce manuel que pour les nombreuses corrections que nous avons apportées au cours de la rédaction.

Sa maîtrise du langage SCRIPT de CMS nous a permis de lui laisser l'entière responsabilité de la mise en page. Nous ne pouvons évaluer combien cet état de fait nous a aidés dans la réalisation définitive de ce cours.

Nous remercions aussi le service de reproduction du C.I.C.G. auquel nous devons la réalisation matérielle de ce manuel.

AVERTISSEMENT

Le but de ce manuel est de servir de support à un cours sur le système CP67.

Les principes et les algorithmes exposés ici sont, dans leur quasi-totalité identiques à ceux du nouveau système VM sur IBM/370, annoncé le 2 août 1972.

C H A P I T R E I



INTRODUCTION

Au cours des vingt dernières années, le mode d'utilisation des ordinateurs a évolué en fonction de l'augmentation de leur puissance et de leur complexité. Cette croissance s'est développée sous les pressions conjuguées et imbriquées des utilisateurs, des progrès de la technologie, et des nombreuses recherches effectuées en vue d'une utilisation optimale des ressources d'une installation.

Dans les années 1950, la programmation est affaire de spécialistes car les ordinateurs ne possèdent aucun support de programmation. L'utilisateur dispose de toute la machine, et s'occupe lui-même des diverses manipulations pour obtenir ses résultats. Cette méthode présente un avantage certain.

En effet la mise au point du programme s'effectue sur la machine par interaction entre l'utilisateur et l'ordinateur. Les dispositifs de mise au point offrent l'exécution instruction par instruction du programme, l'arrêt sur adresse, la visualisation sur oscilloscope ou voyants lumineux du contenu de la mémoire centrale et le rangement manuel d'informations dans la mémoire. Parfois, l'utilisateur dispose d'un programme qui interprète chaque instruction et en imprime le résultat; c'est une première aide à la programmation.

Cette façon de travailler présente aussi un inconvénient majeur: la programmation et la mise au point des programmes étant très proches de la logique interne du calculateur, le rendement global du matériel est fonction de la parfaite connaissance de la machine par l'utilisateur; le rapport entre la vitesse de fonctionnement interne de la machine et les réactions humaines conduit de toutes façons à un rendement faible.

En l'espace de dix ans environ, vers les années 1960, le nombre croissant d'utilisateurs non spécialistes entraîne un développement rapide des outils de programmation (compilateurs, chargeurs ...). Parallèlement, les progrès de la technologie permettent d'augmenter la puissance de traitement des ordinateurs. D'un point de vue économique, pour utiliser au mieux ce potentiel de puissance, il devient

nécessaire de minimiser l'intervention humaine en créant un mécanisme d'enchaînement automatique des travaux sur la machine. C'est l'apparition des systèmes d'exploitation dits "Batch Processing", organisés autour d'un programme moniteur résidant en mémoire. Avec ce nouveau mode de fonctionnement l'exploitation de l'ordinateur est confiée à un opérateur non spécialiste en programmation qui utilise le matériel suivant un mode d'emploi précis.

Le programmeur ne manipule plus lui-même la machine, il doit déposer ses programmes dans un casier d'entrée, et il retrouve ses résultats dans un casier de sortie. Il travaille avec des langages d'assemblage ou de haut niveau; il accède aux unités périphériques à l'aide des programmes généraux du moniteur qui permettent d'utiliser de façon efficace les dispositifs d'entrée-sortie de plus en plus complexes (simultanéité entre les canaux et l'unité centrale).

Du côté machine, la mise en oeuvre d'un tel mode de fonctionnement entraîne l'apparition de notions nouvelles; il devient indispensable:

- d'empêcher le programme utilisateur de détruire le moniteur. Pour cela, le système emploie le mécanisme de protection mémoire.
- de permettre au moniteur, grâce au mécanisme

d'interruptions, de reprendre le contrôle en cas d'incidents de fonctionnement du travail en cours. Le moniteur abandonne alors ce travail et initialise le travail suivant.

- d'arrêter automatiquement un travail au bout d'un temps déterminé, à l'aide d'une horloge interne. On résout ainsi le problème posé par un programme qui "tourne en rond".

Les perfectionnements ultérieurs des systèmes "Batch Processing" n'amènent guère de changements dans la façon de travailler des utilisateurs, si ce n'est une plus grande souplesse d'emploi au prix d'une plus grande complexité des langages de commandes. Cependant des techniques nouvelles, en particulier la multiprogrammation, ont considérablement amélioré le rendement global des installations; en contrepartie elles ont généralement donné naissance à des systèmes extrêmement complexes.

Actuellement l'utilisateur d'un système "Batch Processing" bénéficie d'un large éventail de possibilités nouvelles. Par contre, il a perdu tout moyen d'interaction directe avec son programme.

Pour rétablir cette interaction, des systèmes d'un type nouveau ont été développés ces dernières années et

répondent à des besoins spécifiques ou à des besoins plus généraux. Ces systèmes donnent à chaque utilisateur la possibilité de rester en liaison directe avec son programme.

Pour ce faire, on a imaginé de remplacer les organes traditionnels d'entrée-sortie (lecteurs de cartes et imprimantes) par des machines à écrire connectées à l'ordinateur par ligne téléphonique, et, compte tenu de la lenteur relative des réactions humaines par rapport à la vitesse interne des machines, d'affecter successivement l'unité centrale, par tranches de temps, à chaque utilisateur, donnant ainsi à celui-ci l'impression de disposer de toute la machine.

Alors que les systèmes orientés vers des applications particulières ont, en général, donné satisfaction, les systèmes polyvalents sont devenus lourds et compliqués et n'ont pas eu les performances escomptées. Ces essais ont cependant permis de dégager certaines idées fondamentales, telles que le temps partagé, la mémoire virtuelle, les composants conversationnels. Par la suite, des systèmes moins ambitieux, donc plus simples, ont vu le jour et leurs performances sont acceptables.

Actuellement les systèmes en temps partagé offrent de

grandes facilités à leurs utilisateurs; ils imposent cependant, de par leur conception, des liens étroits entre programmes et système. En particulier, ils ne donnent accès aux possibilités de la machine que par le biais de leurs conventions; ceci exclut de les utiliser pour mettre au point de nouveaux systèmes, pouvant fonctionner sans modifications sur la même machine.

Une approche originale de ce problème a été définie et réalisée en 1967 par le Centre Scientifique IBM de Cambridge (Massachussets - U.S.A.) qui a développé le système CP/67.

INTRODUCTION AU SYSTEME CP-67.

Le programme de contrôle CP-67, actif sur un calculateur IBM-360/67, est un générateur de machines virtuelles, dans un contexte de temps partagé et de multiprogrammation. Il transforme le calculateur réel sur lequel il est actif en plusieurs machines virtuelles indépendantes les unes des autres. Celles-ci diffèrent entre elles par leurs caractéristiques (taille mémoire, unités d'entrée-sortie, etc...); elles sont toutefois la réplique de machines pouvant réellement exister. De façon plus formelle appelons élément virtuel la simulation "software" de son équivalent physique (unité centrale, mémoire, unités d'entrées-sorties); par définition, une machine virtuelle est un ensemble d'éléments virtuels qui constituent sa configuration.

Un aspect fondamental du principe des machines virtuelles est le suivant: tout programme autonome écrit sur une machine virtuelle, de configuration donnée, peut aussi fonctionner sans modification sur une machine réelle de même configuration. D'une façon générale, CP-67 tel qu'il est actuellement, réalise correctement sa tâche de simulation de

machines réelles. Il suffit pour s'en convaincre d'examiner la diversité des systèmes qui ont pu être activés sur une machine virtuelle: TSS360, MTS, OS (PCP, MFT, MVT) DOS, APL, CMS, etc... et CP-67 lui-même.

On peut remarquer cependant que certaines limitations, dues au principe même du temps partagé, font que la plupart des applications "temps réel" ne peuvent être réalisées de façon satisfaisante.

Lorsqu'un utilisateur, par l'intermédiaire d'un terminal connecté au 360/67, se fait reconnaître de CP grâce au nom et au mot de passe de sa machine virtuelle, CP construit la configuration virtuelle correspondante. Pour accéder à sa machine virtuelle, l'utilisateur dispose alors de ce terminal sur lequel CP simule fonctionnellement le pupitre d'un 360 réel. Par exemple, grâce à la fonction IPL de CP, on peut simuler le bouton LOAD du pupitre et initialiser ainsi le système de son choix. D'autre part ce même terminal représente pour la machine virtuelle la console maîtresse d'un 360 réel, console qui est en général utilisée par les systèmes comme console opérateur et, qui, rappelons-le, est une unité d'entrée-sortie comme les autres.