

Architecture du Système d'Exploitation

Claude CARRÉ

Table des matières

	Page
Le contexte	3
Concepts fondamentaux	6
Espace d'adresse	8
Protection	10
Interactif et transactionnel	12
Outils de développement du logiciel	14
Maintenabilité et opérabilité	16
Dates clés du système GCOS 64 / GCOS 7	18
Conclusion	20

Le contexte (1/3)

- Demandes initiales du marketing en 1970 :
 - Reprise petits systèmes Bull-GE (Gamma 100) et Honeywell (H 200/2000).
 - **Système moyen** (*prix de la config. moyenne : 100 K\$*) placé entre les « grands systèmes » sous responsabilité de Phoenix et les petits systèmes sous responsabilité italienne.
 - **LA** référence commerciale est IBM (pressions pour ré-écrire un "IBM/DOS").
 - Le système doit se limiter au traitement du "batch".

- Coût élevé des ressources matériel :
 - Taille min. de la mémoire centrale : 192 K octets; taille max. permise par le matériel : 1 M octets.
 - Nombre moyen de disques : entre 2 et 3 disques amovibles; capacité de chaque disque : 17 M octets.

Le contexte (2/3)

- Préservation de l'avenir en anticipant les évolutions inéluctables :
 - Expériences Multics et Burroughs.
 - Disponibilité, sécurité et maintenabilité accrues ("THE operating system" de Dijkstra, niveaux d'abstraction).
 - Mesures conservatoires pour que le système puisse naturellement (c'est à dire sans modification) supporter des matériels multiprocesseurs et offrir le temps partagé (time sharing).
 - Mécanisme de mémoire virtuelle évitant aux programmeurs la déclaration fastidieuse "d'overlays" et permettant l'adaptation automatique à la taille de mémoire disponible.
 - Mixage aisé de langages de programmation différents.

Le contexte (3/3)

- Le résultat fut la recherche du meilleur compromis entre les demandes marketing, l'anticipation du futur et les contraintes de coût du matériel :
 - Architectures matériel et logiciel définies en étroite coordination.
 - On n'a pas ré-écrit un DOS.
 - Majeure partie du système écrite en langage de haut niveau.
 - Mécanisme de mémoire virtuelle (segmentation) faisant correspondre les unités gérées avec le découpage des programmes tel que vu par le programmeur (contrairement à la pagination).
 - Ressources (matériel & logiciel) nécessaires à l'exécution des programmes déclarées au préalable à la construction des programmes ou à leur exécution.

Concepts fondamentaux (1/2)

- Pour permettre l'implémentation de "serveurs logiciels" (émulateurs, sous-systèmes transactionnels, gestionnaires de périphériques), un programme est défini comme un ensemble de N processus (séquence linéaire d'instructions), appelé GROUPE DE PROCESSUS (défini à l'édition de liens).
- L'exécution de la plupart des programmes courants (compilateurs, éditeurs, etc.) se compose le plus souvent d'un PROCESSUS unique (N=1).
- Toute instruction en exécution fait partie d'un processus et les processus communiquent exclusivement par SEMAPHORES.
- Le système d'exploitation lui-même est un groupe de processus.

Concepts fondamentaux (2/2)

- On associe un ensemble de ressources à un groupe de processus :
 - Un ensemble d'unités de compilation ("Compile Unit") défini à la compilation et à l'édition de liens qui forme son espace d'adresse.
 - Un ensemble de fichiers défini à l'exécution ("Job Control Language") ainsi que le nombre et la taille des tampons de mémoire centrale nécessaires à l'utilisation de ces fichiers.
 - La taille de mémoire centrale recommandée pour l'exécution ("Working Set").
- Une suite de commandes JCL interprété avant exécution définit la séquence et l'ordonnancement des programmes (groupes de processus) à exécuter).

Espace d'adresse (1/2)

- Toute instruction et toute donnée appartiennent à un segment.
- Les compilateurs ne génèrent que des segments d'instructions "purs": les programmes ne peuvent s'autodétruire, sont réentrants et partageables entre processus

Espace d'adresse (2/2)

- Les descripteurs de segments sont regroupés en tables; chaque table a un type :
 - Type 3 : Espace d'adresse propre à un processus (code et données).
 - Type 2 : Segments partagés entre tous les processus d'un groupe, exemple : les procédures communes à tous les processus d'un gestionnaire d'imprimantes.
 - Types 1 & 0 : Segments partagés entre tous les processus de tous les programmes ou tous les processus d'un sous-système transactionnel TDS :
 - ✓ Les segments de type 0 sont des procédures et tables du système d'exploitation, par exemple les méthodes d'accès aux fichiers (UFAS), ce qui permet une multiprogrammation de niveau automatiquement adapté à la charge instantanée du système (sans avoir à configurer le nombre de processus du système).
 - ✓ Les segments de type 1 peuvent être attachés et détachés dynamiquement, par exemple le traitement des transactions d'un sous-système transactionnel TDS (les "Transaction Processing Routines" ou TPR).

Protection (1/2)

■ Haute protection offerte par la segmentation :

- Les dégâts provoqués par un programme qui n'est pas au point sont limités : pas de branchement possible dans un autre programme, pas de modification des constantes ou du programme en cours d'exécution.
- La pagination des segments prévue dès l'origine a été introduite dans les années 80. Elle fut uniquement destinée à permettre une gestion plus simple de la mémoire.

Protection (2/2)

- Utilisation des 4 niveaux de privilège ("rings") :
 - Utilisation la plus limitée possible du niveau le plus privilégié (ring 0) réservé aux quelques procédures du système qui le requièrent.
 - Appels au système d'exploitation vers des entrées bien définies, par appel de procédure standard avec transition du niveau de privilège ("ring crossing").
- Protection du système de fichiers non intégrée au système de base (à la différence de Multics) :
 - en raison de l'amovibilité des disques,
 - et de la compatibilité (formatage physique et formats des fichiers) IBM.

Interactif et transactionnel (1/2)

- Le transactionnel est rapidement apparu comme une nécessité commerciale (compétence distinctive des grands systèmes Bull et des équipes de Phoenix) :
 - Un système transactionnel est "fermé" : ses utilisateurs n'accèdent qu'à un nombre défini de commandes et de fichiers.
 - Le nombre d'utilisateurs simultanés peut être très élevé (plusieurs centaines, voire milliers).
 - Il est généré comme un groupe de processus banalisé ("TDS") dont le nombre est fonction du nombre d'utilisateurs prévus.
 - En cas de grand nombre de transactions, elles sont implantées dans des segments de type 1.
 - L'intégrité des bases de données et la haute disponibilité sont indispensables et sont offertes par les "journaux" et la gestion des verrous ("Generalized Access Control", GAC).

Interactif et transactionnel (2/2)

- Le temps partagé ou time sharing (nom commercial : "Interactive Operator Facility, IOF") correspond à une utilisation non prévisible des ressources programmes et fichiers du système par un opérateur sur terminal :
 - À chaque commande d'un utilisateur sur un terminal correspond un programme monoprocessus.
 - C'est une extension naturelle du système qui ne nécessite aucun redesign.

Outils de développement du logiciel (1/2)

- Alias "Software Factory".
- Langage de développement de haut niveau envisagé dès 1968 ("Q-Language") remplacé par un sous-ensemble de PL/1 en 1970).
- Le développement du logiciel à partir de 1971 en parallèle avec celui du matériel nécessita la réalisation d'outils sur des systèmes Multics (non commercialisés à l'époque) :
 - un système d'abord à Boston,
 - puis un système (constitué de pièces de récupération) à Paris qui supportait moins de 10 utilisateurs simultanés (hors simulation, interdite).
- Connexion de terminaux de Paris sur le système de Boston par ligne spécialisée uniquement le matin (décalage horaire oblige).

Outils de développement du logiciel (2/2)

- Outils sur Multics : Compilateur SHPL, assembleur SNAL, system linker et simulateur d'instructions (hors entrées / sorties), tous développés à Boston.
- Référence du code source à Boston (transferts par cartes perforées, puis par bandes).
- La mise au point du matériel utilise un petit système d'exploitation (SCP) qui se "bootstrappe" (s'amorce) à partir de cartes perforées.
- Le système GCOS 64 sera ensuite construit par remplacement des modules de SCP par ceux de GCOS développés sur Multics.
- La transition SHPL vers HPL et SNAL vers NAL a constitué un jalon important dans l'histoire de GCOS.

Maintenabilité et opérabilité (1/2)

- Système voulu "fermé" vis-à-vis des clients :
 - Langages d'implémentation du système et interfaces d'accès au système non livrés en clientèle.
 - Support des grands clients par équipe spécialisée créée en 1977 avec livraison de :
 - ✓ un sous-ensemble du langage d'implémentation HPL : GPL,
 - ✓ un sous-ensemble des interfaces système,
 - ✓ "Exits user".
- La reproductibilité d'un problème client sur les systèmes du constructeur est facilitée par :
 - L'absence de génération du système en clientèle (livraison d'une image binaire unique).
 - la limitation du nombre de versions en clientèle.
- La correction des problèmes est facilitée par l'implémentation du système en langage de haut niveau.

Maintenabilité et opérabilité (2/2)

- Génération du système d'exploitation comme n'importe quel programme multiprocessus.
- Souci de l'amélioration de la visibilité externe du système et de son opérabilité :
 - Équipe spécialisée mise en place dès les années 1975 qui se met en relation avec des ergonomes de l'INRIA.
 - Nouveau langage de commande GCL livré en 1984.

Dates clés du système GCOS 64 / GCOS 7 (1/2)

Avril 1974	Mode G100, puis H200 Pas d'application native Utilitaires natifs de gestion de périphériques
1976	Mode natif batch : Programmes COBOL (et FORTRAN) Fichiers compatibles IBM
1977	Mode transactionnel TDS Fichiers UFAS et bases de données IDS 2
1978	Mode interactif IOF
1979	Modes SIRIS 8 et SIRIS 3 Support de frontaux télécom Datanet Accès aux réseaux X25

Dates clés du système GCOS 64 / GCOS 7 (2/2)

1984	Support de matériels multiprocesseurs symétriques Nouveau langage de commande GCL Bases de données Oracle
1986	Gestion de mémoire virtuelle paginée
1990	Support de périphériques (disques, cartouches) par protocole SCSI Bus d'entrées/sorties standard Multibus II 200 utilisateurs IOF
1991	Support des disques miroirs Sous-système UNIX (accès aux réseaux TCP/IP) Mémoire centrale : 256 MO
1994	Support d'hexaprocesseurs 7000 utilisateurs TDS
2001	GCOS 7 sur processeur standard INTEL

Conclusion

- En partant d'objectifs très modestes, l'architecture définie il y a près de 35 ans fut suffisamment extensible pour satisfaire des clients actuels aux besoins importants.

1974 : 1ère annonce	2004
Système monoprocesseur	De 1 à 24 processeurs
Mémoire centrale de 192 KO à 640 KO	Mémoire centrale de 8 GB (x 40 000)
Connexion de 2 à 24 disques de 17 MO, soit un total de 0,4 GO	Connexion de plus de 100 disques, soit un total de 1500 GO (x 40 000)
Puissance CPU référence 1	Puissance CPU référence 3200
Pas d'utilisateur transactionnel	7000 utilisateurs transactionnels