Atlas Supervisor: le premier système d'exploitation moderne

Stéphane FOSSE
fosse.fr

29 décembre 2025

Au début des années 1960, programmer un ordinateur ressemble à conduire un véhicule en construisant simultanément la route devant soi. L'ordinateur Elliott 803B de l'époque charge un compilateur Algol 60 dans une mémoire vide et s'exécute directement sur le matériel nu. Aucune couche logicielle n'existe entre le programme et la machine. Cette simplicité cache une limite fondamentale : lorsque Manchester University et Ferranti Limited développent Atlas, un ordinateur cent fois plus rapide que tout ce qui existe alors, cette approche montre ses failles. Comment gérer efficacement une machine aussi puissante sans qu'elle attende constamment que les périphériques lents terminent leurs opérations ?

La question se pose avec acuité. Les bandes magnétiques transfèrent 100 000 caractères par seconde, les lecteurs de cartes perforées à peine 1 000. Entre ces deux mondes, l'unité centrale peut théoriquement traiter plus d'un million de caractères. Atlas passerait l'essentiel de son temps à attendre. Tom Kilburn et son équipe à l'université de Manchester comprennent qu'une nouvelle approche s'impose. Ils inventent Atlas Supervisor, un programme qui contrôle toutes les activités du système et orchestre l'utilisation des ressources. Per Brinch Hansen qualifiera plus tard cette réalisation de « percée la plus significative de l'histoire des systèmes d'exploitation ».

Une architecture de protection matérielle et logicielle

Le problème de la protection se révèle central. Comment permettre à plusieurs programmes de partager le même ordinateur sans qu'ils interfèrent entre eux ou avec le superviseur ? L'équipe conçoit trois registres de contrôle distincts : *Main Control* pour les programmes utilisateurs, *Extracode Control* pour les sous-routines et le superviseur, *Interrupt Control* pour les routines de gestion des périphériques. Lorsqu'un programme s'exécute sous Main Control, l'accès à la mémoire auxiliaire et au V-store, ensemble de bascules contrôlant les périphériques, lui est interdit par le matériel. Toute tentative génère une interruption et bascule vers le superviseur.

Cette séparation stricte des privilèges s'accompagne d'un mécanisme de verrouillage de pages mémoire. Chaque bloc de 512 mots dans la mémoire centrale dispose d'un registre d'adresse associé et d'un bit de verrouillage. Le superviseur peut interdire l'accès à certaines pages aux programmes utilisateurs tout en y accédant lui-même sous Interrupt Control. Les routines d'interruption utilisent uniquement certains registres d'index, les B-lines 111 à 118, jamais touchés par les programmes. Cette discipline rend inutile la sauvegarde des registres lors d'une interruption. Le basculement prend quelques microsecondes.

David Howarth rejoint Ferranti alors que le développement bat son plein. Son patron lui suggère de travailler sur le « compilateur-compilateur » de Tony Brooker. Howarth comprend rapidement que la vraie valeur d'Atlas réside ailleurs. Il se concentre sur le superviseur. Son collègue John Crowther raconte plus tard que les contributions d'Howarth fonctionnaient à 90% dès la première tentative. Les autres membres de l'équipe dépassaient rarement 60%. Cette fiabilité impressionne d'autant plus que le code s'écrit pratiquement en langage machine, sans possibilité réelle de test préalable.

Le superviseur s'organise en branches dormantes qui s'activent à la demande. Les routines d'interruption forment la couche la plus fréquemment sollicitée. Lorsqu'un lecteur de bande papier remplit son buffer d'un caractère, il positionne un flip-flop dans le V-store. Si les interruptions ne sont pas inhibées, le contrôle bascule vers Interrupt Control et un programme en mémoire fixe détermine la cause. Un matériel spécial attaché au registre B123 identifie la source en deux à six instructions. La priorité intégrée correspond à l'urgence de chaque équipement. Les transferts caractère par caractère se produisent à haute fréquence. Le lecteur de bande fonctionne à 300 caractères par seconde. L'imprimante génère 50 interruptions par révolution, une toutes les 1,5 millisecondes.

Certaines situations demandent des séquences plus longues que ce que permettent les routines d'interruption. Lorsqu'un lecteur de bande termine la lecture d'un rouleau complet, le système doit traiter les caractères collectés, les écrire sur bande magnétique si nécessaire, décoder et lister les titres. La routine d'interruption initie alors un

Supervisor Extracode Routine. Ces S.E.R. constituent les branches principales du superviseur. Elles s'activent soit par des routines d'interruption, soit par des instructions extracode dans un programme utilisateur. Elles utilisent la mémoire auxiliaire comme espace de travail et des zones de mémoire centrale et tambour verrouillées. Elles opèrent sous Extracode Control après avoir préservé le registre de contrôle extracode du programme en cours.

Le coordinateur et la gestion du temps partagé

Plusieurs S.E.R. peuvent s'activer simultanément. Une routine spéciale en mémoire fixe, le coordinateur, organise leur exécution en série. Les S.E.R. en attente se rangent dans quatre files de priorité différentes. La file de priorité maximale contient les routines liées aux transferts tambour et aux erreurs matérielles critiques comme les erreurs de parité mémoire. La deuxième file concerne les bandes magnétiques, la troisième les périphériques. La dernière file contient une entrée par programme en cours d'exécution. Lorsqu'un S.E.R. se termine ou se suspend, le coordinateur sélectionne le premier S.E.R. activé dans la file de priorité la plus élevée.

Un S.E.R. peut se suspendre volontairement. S'il demande le transfert d'un bloc depuis le tambour vers la mémoire centrale, il se met en pause jusqu'à la fin du transfert. La file des requêtes de transfert tambour, maintenue dans la mémoire auxiliaire, peut saturer. L'S.E.R. demandeur se suspend alors. Avant de se suspendre, il spécifie un point de reprise. Un S.E.R. suspendu redevient libre de continuer quand la cause de la suspension disparaît. Le S.E.R. qui contrôle les transferts tambour et extrait les entrées de la file réveille les routines suspendues. Les files peuvent donc contenir à tout moment des routines en attente d'exécution et des routines suspendues. Le nombre total reste limité : un par programme utilisateur, un ou deux par source d'interruption selon les caractéristiques de chaque routine.

Les programmes utilisateurs s'insèrent dans cette architecture comme des branches de priorité inférieure aux S.E.R. Bien que logiquement sous-programmes du superviseur, ils peuvent fonctionner longtemps en utilisant pleinement les ressources sans référence au superviseur. Quand ils demandent une action contrôlée par le superviseur, comme un transfert d'entrée-sortie, ils utilisent une des 512 instructions extracode possibles. Ces instructions basculent vers Extracode Control et entrent dans la mémoire fixe à l'un des 512 points d'entrée prédéfinis. Cette limitation du nombre de points d'entrée garantit une protection complète. Les programmes utilisent des numéros logiques pour désigner les périphériques et les blocs de stockage. Le superviseur établit la correspondance avec les ressources physiques réelles.

Le changement de contexte entre programmes illustre l'équilibre recherché entre souplesse et performance. Basculer d'un programme à un S.E.R. et retour prend environ 40 instructions. Changer de programme utilisateur demande davantage : environ 750 plus 12 fois le nombre de pages mémoire. Sur l'Atlas de Manchester avec 32 pages de mémoire centrale, l'aller-retour complet pour changer de programme puis revenir prend 2,5 millisecondes. Ce coût pousse vers une stratégie claire : minimiser les changements de programmes, maximiser l'utilisation du basculement rapide vers les routines d'interruption et les S.E.R.

L'invention du store à un niveau change la donne. La mémoire centrale se divise en pages de 512 mots. Le tambour utilise la même taille de blocs, appelés secteurs. Un registre d'adresse de page associé à chaque page contient les bits d'adresse les plus significatifs du bloc d'information occupant la page. Quand le matériel doit accéder à un mot en mémoire centrale, il cherche l'équivalence entre l'adresse de bloc demandée et le contenu de chaque registre d'adresse de page en parallèle. L'absence d'équivalence provoque une interruption de non-équivalence. Le superviseur maintient dans la mémoire auxiliaire un répertoire de blocs listant l'emplacement de chaque bloc d'information : soit une page de mémoire centrale, soit un secteur du tambour. Une seule copie existe à un moment donné.

Quand un programme demande un bloc absent de la mémoire centrale, l'interruption de non-équivalence active le superviseur. Il transfère le nouveau bloc depuis un secteur du tambour vers une page de la mémoire centrale. Le superviseur maintient toujours une page vide disponible pour recevoir immédiatement un bloc depuis le tambour. Pendant ce transfert, il prépare le transfert d'une autre page vers le tambour pour maintenir une page vide. Le programme d'apprentissage sélectionne la page à écrire. Il analyse l'utilisation passée des blocs pour déterminer la page contenant le bloc le moins susceptible d'être requis prochainement. Il intègre une rétroaction : s'il écrit un bloc qui revient presque immédiatement, il ne répète pas l'erreur.

Ce système de store combiné cache au programmeur la nature à deux niveaux du stockage. Tous les programmes utilisent ce concept, y compris les routines du superviseur lui-même. Beaucoup de routines du superviseur, peu fréquemment utilisées, occupent normalement des secteurs sur le tambour. Elles migrent vers la mémoire centrale quand nécessaire, sans perturber les programmes utilisateurs sauf par une possible augmentation temporaire des transferts tambour. La facilité avec laquelle les blocs individuels se déplacent dans le store combiné devient inestimable pour le fonctionnement d'un système d'exploitation efficace. Tom Kilburn et son équipe considèrent plus tard que gérer efficacement un grand ordinateur sans cette facilité paraît presque impossible.

Le système d'exploitation organise le flux des travaux à travers l'ordinateur. Chaque travail nécessite plusieurs documents d'entrée. Une description de travail précède les données. Elle liste les titres de tous les documents requis, les flux de sortie produits, les bandes magnétiques nécessaires, les limites supérieures d'espace de stockage

et de temps de calcul. Le superviseur assemble les travaux complets dans le puits d'entrée avant l'exécution. Les documents d'entrée se copient sur une bande magnétique système au fur et à mesure de leur réception. Si l'espace dans le store principal devient insuffisant, ils se récupèrent depuis cette bande quand le travail est prêt à s'exécuter. Le puits d'entrée se divise en deux parties : le puits d'entrée A contient les documents attendant d'autres documents avant utilisation, le puits d'entrée B contient les ensembles complets de documents pour les travaux.

La bande système d'entrée joue un rôle similaire à celui d'un système conventionnel, mais l'ordinateur la prépare lui-même au lieu d'équipements hors ligne. Aucune manipulation de bande ni supervision manuelle n'est requise après l'entrée des documents originaux. Un point décisif pour un système conçu pour traiter beaucoup de travaux divers. La même bande sert à écrire des blocs d'entrée en séquence consécutive et à relire des blocs précédemment écrits pour récupérer des documents particuliers quand requis. La bande effectue donc des balayages fréquents sur quelques mètres, tout en progressant graduellement vers l'avant. La longueur de ces balayages dépend de l'espace du store principal occupé par le puits d'entrée A. Tant que les balayages ne dépassent pas environ 80 pieds soit 130 blocs, le temps d'attente pour écrire de nouveaux blocs reste inférieur au temps d'entrée de trois blocs depuis un lecteur de cartes.

Un puits de sortie fonctionne de manière analogue. Une bande système de sortie assure le tampon en masse. La sortie pour tous les périphériques se place sur la même bande, organisée en sections subdivisées de sorte que le contenu d'une section occupe tous les périphériques actuellement en fonctionnement pendant la même durée. Si une rafale de sortie se génère pour un périphérique particulier, elle s'espace sur la bande, laissant des blocs libres à remplir plus tard avec la sortie pour d'autres périphériques. La récupération d'information depuis la bande vers le puits de sortie B consiste simplement à lire des sections complètes. Un S.E.R. surveille la quantité d'information restant dans le puits de sortie B pour chaque équipement et la relie à la distance de balayage présente pour décider quand commencer à déplacer la bande en arrière pour la prochaine opération de lecture.

Une troisième bande magnétique système, la bande de vidage, traite les situations d'urgence. Quand les demandes de stockage dépassent la capacité du store principal et des bandes d'entrée-sortie, cette bande conserve l'information non immédiatement requise. L'exécution d'un problème peut se suspendre et le problème s'enregistrer temporairement sur la bande de vidage si d'autres problèmes doivent remplir le puits de sortie, ou si sa propre sortie ne peut se loger dans le puits de sortie. Les puits d'entrée et de sortie peuvent déborder vers la bande de vidage. Cette bande ne s'utilise pas de manière systématique mais pour gérer les urgences. Le système permet même de se passer des bandes système d'entrée et de sortie si nécessaire, réduisant les puits et augmentant la charge sur la bande de vidage. Dans un cas extrême, la bande de vidage elle-même peut disparaître, impliquant une réduction supplémentaire de l'efficacité du système.

La machine entre en service en 1962. Elle marque une rupture. Avant Atlas, compiler ou exécuter un programme impliquait un long processus manuel. L'opérateur chargeait le compilateur, les données, surveillait l'exécution, récupérait les résultats. Chaque étape nécessitait une intervention humaine. Atlas automatise cette chaîne. Les documents arrivent dans n'importe quel ordre par n'importe quel périphérique. Les titres et descriptions de travail permettent au superviseur d'assembler et d'exécuter des programmes complets. La sortie se distribue sur tous les périphériques disponibles. Les programmes se compilent et s'exécutent généralement dans l'ordre de complétion de l'entrée, mais le superviseur ajuste selon la charge sur différentes parties du système.

Trois flux de travaux coexistent. Les travaux limités par le calcul utilisent des puits d'entrée-sortie suffisamment grands pour contenir tout leur matériel. Pendant qu'un travail utilise l'unité centrale, l'information concernant d'autres travaux occupe les équipements périphériques. Les travaux limités par les bandes accèdent directement aux bandes magnétiques. Quand un travail attend des transferts de bande, le contrôle bascule vers un travail limité par le calcul pour maintenir l'activité de l'unité centrale. Les longs travaux de calcul forment le troisième flux. Ils peuvent s'interrompre pour exécuter une séquence de courts travaux afin de maintenir des niveaux satisfaisants dans les puits d'entrée-sortie. Sans cette interruption, un long travail empêcherait l'extraction d'information du puits d'entrée ou le passage vers le puits de sortie, provoquant l'arrêt de l'activité périphérique par manque d'espace dans le puits d'entrée et manque de matériel dans le puits de sortie.

Cette architecture pose les fondations de ce que deviendront tous les systèmes d'exploitation modernes. La mémoire virtuelle par pagination à la demande naît avec Atlas. IBM utilisera plus tard le terme Virtual Memory, mais l'équipe d'Atlas parle déjà de one-level store. Le multiprogrammation, cette idée maintenant banale que le système d'exploitation bascule son processeur d'un programme à l'autre si rapidement que tous semblent s'exécuter simultanément, prend forme dans le coordinateur d'Atlas. Les appels système et les appels de bibliothèque via ce qu'Atlas nomme extracodes deviennent standard. Le spooling des données depuis les périphériques lents vers les rapides avant d'exécuter un travail structure le flux d'information.

La structure décrite dans les articles de Kilburn, Payne et Howarth publiés fin 1961 prouve son efficacité. Elle s'adapte à chaque tâche supervisée envisagée. Les variations futures s'intègrent dans cette structure. Son succès tient largement à certaines caractéristiques matérielles d'Atlas, particulièrement les dispositions pour protéger les programmes des interférences et les registres d'adresse de page. La manière dont ces derniers permettent de déplacer l'information pendant l'exécution sans recompiler les programmes donne au superviseur un degré de liberté entièrement nouveau. Les usages possibles de cette facilité se révèlent si étendus que gérer un grand

ordinateur efficacement sans elle semble par comparaison presque impossible. David Howarth et son équipe créent quelque chose qui dépasse largement la somme des solutions techniques. Ils inventent une nouvelle façon de penser l'interaction entre matériel et logiciel, entre performance et protection, entre automatisation et contrôle.

Références

[1] I. Cottam. Heroes of Software Engineering – David Howarth – Atlas Supervised! Software Sustainability Institute, oct. 2013.

[2] D. Howarth. The Atlas Supervisor Program. In : *Proceedings of a Symposium held at the London School of Economics*. Juill. 1962.

[3] T. Kilburn, R. B. Payne et D. J. Howarth. The Atlas Supervisor. In : *AFIPS '61 (Eastern): Proceedings of the December 12–14, 1961, Eastern Joint Computer Conference*. 1961.

[4] The Atlas Supervisor. Computer Museum, University of Stuttgart.