

Etat de l'art des serveurs Unix au dessus de Mach

Rémy Card
card@tsx-11.mit.edu

Franck Mével
mevel@masi.ibp.fr

Julien Simon*
simon@freenix.fr

16 mars 1995

Nous présentons ici brièvement le fonctionnement des systèmes qui offrent une émulation d'Unix au dessus du micronoyau Mach.

Ils sont de trois types :

- les systèmes à architecture monolithique, utilisant Mach comme couches basses (OSF/1 IK) ;
- les systèmes composés d'un seul serveur, qui s'exécute en mode utilisateur au dessus de Mach (BSD Single Server, CMU Unix Server, OSF/1 MK, Lites et Sprite) ;
- les systèmes composés de plusieurs serveurs coopérants, appelés systèmes multiserveurs. (CMU Mach-US et GNU Hurd).

1 Systèmes monolithiques

1.1 OSF/1 IK

OSF/1 IK [Foundation 1990], développé par l'Open Software Foundation, repose sur Mach 2.5. Ce système reprend la structure monolithique présente dans Unix et se contente d'utiliser les fonctionnalités de Mach pour assurer les sémantiques Unix sans en tirer pleinement profit. Pour ces raisons, nous nous contentons de le décrire brièvement.

Sa structure est représentée sur la figure 1.

Les innovations offertes par OSF/1 IK par rapport à Unix, au niveau structurel, sont :

- l'utilisation des fonctionnalités de gestion de tâches et *threads* pour fournir des processus selon la sémantique Unix ;

*This work has not been financed by the MASI Laboratory

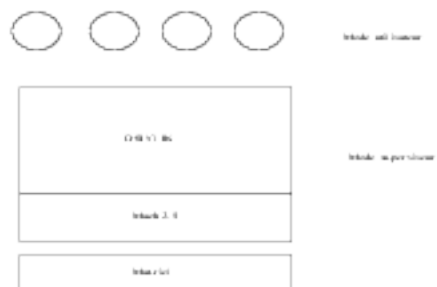


Figure 1: Structure de OSF/1 IK

- la possibilité d'utiliser plusieurs *threads* au sein d'un même processus ;
- la parallélisation des fonctions du système, grâce aux fonctionnalités fournies par Mach, afin de permettre une exécution optimale sur des ordinateurs multiprocesseurs ;
- l'utilisation des gestionnaires de périphériques contenus dans Mach pour effectuer les entrées-sorties ;
- l'optimisation de la gestion de la mémoire virtuelle grâce à l'utilisation des mécanismes de Mach (notamment par rapport à la version BSD de Unix qui implémente une gestion peu performante de la mémoire¹ [Leffler *et al.* 1989]).

L'utilisation de fonctionnalités de Mach permet d'alléger le code du système lui-même puisque de nombreuses fonctions internes (ordonnancement de processus, gestion des périphériques, gestion de la mémoire physique...) peuvent être remplacées par des appels à Mach. Néanmoins, de par son caractère monolithique, OSF/1 IK n'est pas différent d'une implémentation classique de fonctionnalités Unix dans la mesure où il est fondé sur la même structure et en possède donc les mêmes inconvénients.

2 Systèmes monoserveurs

2.1 OSF/1 MK

2.1.1 Structure de OSF/1 MK

OSF/1 MK [Places *et al.* 1992], développé par l'Open Software Foundation, repose sur la version 3.0 de Mach. L'architecture de ce système est directement inspirée de celle du BSD Single Server de Carnegie Mellon University. C'est également le cas du Unix Server de CMU [Golub *et al.* 1990], et de Lites [Helander 1994].

Il est donc composé d'un serveur unique, qui s'exécute en mode utilisateur au dessus de Mach. Les composantes de ce serveur sont :

- la quasi-intégralité du code de OSF/1 IK ;
- une couche d'interface avec les processus utilisateurs, chargée de convertir leurs appels système en RPC à destination de Mach ;
- une couche d'interface avec le noyau, qui permet au serveur d'obtenir des services indisponibles en mode utilisateur.

Sa structure est représentée sur la figure 2.

¹ pour les versions antérieures à 4.4BSD, cette dernière ayant repris la gestion de la mémoire virtuelle de Mach

2.1.2 Traitement des appels système

OSF/1 MK offre une compatibilité binaire avec Unix, grâce à une librairie d'émulation placée dans l'espace d'adressage des processus utilisateurs.

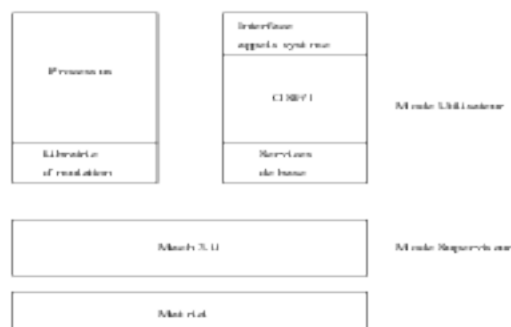


Figure 2: Structure de OSF/1 MK

Lorsqu'un processus effectue un appel système, il exécute une trappe qui transfère le contrôle à Mach. Celui-ci redirige l'appel vers la librairie, qui exécute l'appel système. Si l'appel est simple, la librairie peut l'exécuter seule.

Par contre, dans le cas où l'appel est plus complexe, elle effectue un RPC vers le micro-noyau et retourne les résultats au processus utilisateur.

Cependant, l'utilisation d'une librairie d'émulation pose de sérieux problèmes de sécurité et de maintenance. Pour tenter de les résoudre, OSF a mis au point une version d'OSF/1 MK sans librairie d'émulation [Patience 1993]. Ceci a été rendu possible au prix de plusieurs modifications de Mach 3.0, notamment dans la gestion de la mémoire virtuelle et des exceptions.

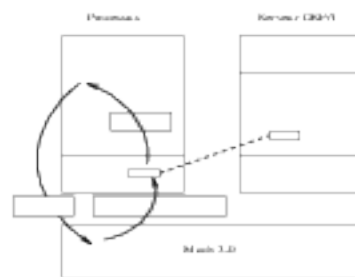


Figure 3: Appel système simple dans OSF/1 MK

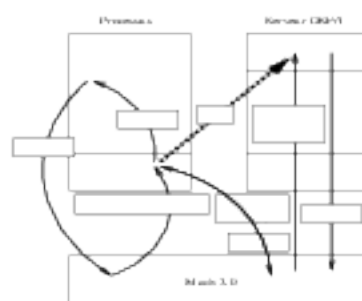


Figure 4: Appel système complexe dans OSF/1 MK

2.1.3 Gestion des processus

Les processus sont gérés par le biais des tâches et threads Mach. Le serveur maintient des structures de données contenant les informations liées aux sémantiques Unix des processus qui ne sont pas directement gérées par Mach. Les informations maintenues par le serveur sont également accessibles par la librairie d'émulation sans son espace d'adressage, afin de lui permettre de traiter directement certains appels système, sans faire appel au serveur.

La mémoire virtuelle des processus est gérée par un paginateur, appelé "vnode pager", qui fait partie du serveur.

2.1.4 Gestion des entrées-sorties

Le serveur contient une version non modifiée des fonctions d'entrées-sorties d'Unix BSD (systèmes de fichiers, gestion des sockets, ...). Dans Unix, ces sous-systèmes utilisent des tables de pointeurs sur fonctions pour accéder au gestionnaires de périphériques [Leffler *et al* 1989].

Dans OSF/1 MK, ces tables contiennent les adresses de fonctions, contenues dans la couche d'interface avec le micronoyau, qui redirigent les requêtes vers les gestionnaires de périphériques contenus dans Mach.

2.2 Sprite

Sprite [Ousterhout *et al* 1988], développé à l'Université de Berkeley, est un système réparti, basé sur les sémantiques Unix. Les fonctionnalités réparties de Sprite permettent à un ensemble de stations reliées par un réseau de partager leurs ressources (processeurs, fichiers) de manière transparente.

Sprite a été porté au dessus de Mach 3.0 [Kupfer 1993], afin de bénéficier du grand nombre de plates-formes supportées par ce micronoyau. Sa structure est représentée sur la figure 5.

Sous cette forme, Sprite s'exécute en tant que serveur unique en mode utilisateur. Si l'objectif de portabilité ait été atteint, les performances sont en revanche très inférieures à celles du système de départ.



Figure 5: Structure de Sprite au dessus de Mach

3 Systemes multiserveurs

3.1 Mach-US

3.1.1 Structure de Mach-US

Développé par Carnegie Mellon University, Mach-US est un système multi-serveurs orienté objet basé sur Mach 3.0 [Guedes et Julin 1991], [Stevenson et Julin 1995]. Les composantes de Mach-US sont :

- un ensemble de serveurs, offrant chacun un type de service bien déterminé ;
- une librairie d'émulation, chargée dans l'espace d'adressage de chaque processus utilisateur, et qui permet d'offrir une compatibilité binaire avec 4.3BSD.

Sa structure est représentée sur la figure 6.

Les serveurs sont de deux types :

- les serveurs indépendants des fonctionnalités système à offrir aux processus utilisateurs :
 - serveur de configuration, chargé de lancer les autres et de leur fournir des informations globales ;
 - serveur d'authentification ;
 - serveur de diagnostic.
- les serveurs offrant des fonctionnalités système aux processus utilisateurs :
 - serveur de chemin ;
 - serveur de fichiers, qui gère les systèmes de fichiers de type UFS ;
 - serveur de communications inter-processus locales (tubes, sockets, ...);
 - serveur de communications réseau par sockets ;
 - serveur de terminaux, qui gère les lignes séries ;
 - serveur de processus.

3.1.2 Traitement des appels système

Les appels système sont implémentés de la même façon que dans OSF/1 MK : les processus utilisateur provoquent une trappe gérée par Mach qui la redirige vers la librairie d'émulation selon un effet de trampoline. Cette librairie exécute ensuite l'appel système en transmettant une ou plusieurs requêtes aux serveurs.



Figure 6: Structure de Mach-US

3.2 GNU Hurd

3.2.1 Structure interne de Hurd

Hurd est en cours de développement par la Free Software Foundation [Bushnell *et al* 1994]. Il est conçu comme un ensemble de serveurs coopérants, auxquels s'ajoute une librairie d'émulation. Certains de ces serveurs sont essentiels au fonctionnement du système alors que d'autres peuvent être remplacés dynamiquement.

Les serveurs indispensables sont :

- le serveur d'authentification ;
- le serveur de processus ;
- les serveurs de fichiers (UFS, NFS, ...);
- le serveur de terminaux.

3.2.2 Mécanisme de translation

La plupart des serveurs sont accessibles via des fichiers. Normalement, l'ouverture d'un fichier provoque l'envoi d'un droit d'émission sur un port caractérisant le fichier. Tout système de fichiers est géré par un serveur ; quand un fichier est ouvert, le serveur crée un port, l'associe au fichier et transmet un droit d'émission sur ce port au client.

Un fichier peut avoir un "translateur" qui lui est associé. Dans ce cas, au lieu de retourner un port caractérisant le fichier, le serveur exécute un programme, appelé "translateur". Le serveur fournit à ce programme le port correspondant au fichier ouvert et le translateur doit renvoyer un port au client.

Ce mécanisme de translation est utilisé pour le montage de systèmes de fichiers : quand un utilisateur ouvre le fichier correspondant à un point de montage, le translateur correspondant, qui comprend la structure physique du système de fichiers monté, est exécuté et renvoie un port au client. Les accès ultérieurs aux fichiers sont donc ensuite dirigés, via ce port, sur ce translateur.

Un translateur peut être associé à un fichier par son propriétaire. De la sorte, tout programme peut être utilisé comme translateur à condition qu'il respecte un protocole bien défini. Il est donc ainsi possible, pour tout utilisateur, d'ajouter dynamiquement des fonctionnalités au système par le biais de translateurs.

Aucun des translateurs présents installé dans le système de fichiers ne voit ses privilèges accrus. Les translateurs s'exécutent avec l'identité et les droits des propriétaires des fichiers auxquels

ils sont associés, et peuvent être associés à des fichiers ou modifiés seulement par leurs propriétaires. Les protocoles d'entrées-sorties et d'accès aux fichiers sont conçus de façon à ne pas faire confiance aux différents clients et serveurs.

3.2.3 Traitement des appels système

Les appels système sont implémentés de la même façon que dans OSF/1 MK : les processus utilisateur provoquent une trappe gérée par Mach qui la redirige vers la librairie d'émulation selon un effet de trampoline. Cette librairie exécute ensuite l'appel système en transmettant une ou plusieurs requêtes aux serveurs.

Références

- [Bushnell *et al.* 1994] M. Bushnell, R. Stallman, R. Mc Grath, et Jan Brittonson. Towards a New Strategy of Os Design. *GNU Bulletin*, January 1994.
- [Foundation 1990] Open Software Foundation. *The Design of the OSF/1 Operating System*. Open Software Foundation., 1990.
- [Golub *et al.* 1990] D. Golub, R. Dean, A. Forin, et R. Rashid. Unix as an Application Program. In *Proceedings of the Summer 1990 USENIX Conference*, pages 87–96, June 1990.
- [Guedes et Julin 1991] P. Guedes et D. Julin. Object-Oriented Interfaxes in the Mach 3.0 Multi-Server System. In *Proceedings of the IEEE Second International Workshop on Object Orientation in Operating Systems*, October 1991.
- [Helander 1994] J. Helander. Unix under Mach - the Lites Server. Master's thesis, Helsinki University of Technology, December 1994.
- [Kupfer 1993] Michael D. Kupfer. Sprite on Mach. In *Proceedings of the Third Usenix Mach Symposium*, April 1993.
- [Leffler *et al.* 1989] S. Leffler, M. Mc Kusick, M. Karels, et J. Quaterman. *The Design and Implementation of the 4.3BSD UNIX Operating System*. Addison-Wesley, 1989.
- [Ousterhout *et al.* 1988] J. Ousterhout, A. Cherenon, F. Douglass, M. Nelson, et B. Welch. The Sprite Network Operating System. *IEEE Computer*, 21(2):23–36, February 1988.
- [Patience 1993] Simon Patience. Redirecting System Calls in Mach 3.0 : an Alternative to the Emulator. In *Proceedings of the Third Usenix Mach Symposium*, April 1993.
- [Places *et al.* 1992] F. Barbou Des Places, P. Bernadat, M. Condict, S. Empereur, J. Febvre, D. George, J. Loveluck, E. McMannis, S. Patience, J. Rogado, et P. Roudaud. *Architecture and Benefits of a Multithreaded OSF/1 Server*. Technical report, OSF Research Institute, June 1992.
- [Stevenson et Julin 1995] J. Stevenson et D. Julin. Mach-Us: Unix on Generic Os Object servers. In *Proceedings of the 1995 Usenix Conference*, January 1995.