INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES CENTRE VAL DE LOIRE

Département Sécurité et Technologies Informatiques

Rapport Ouverture Scientifique et Technique

Auteur

Dubois Louan
Maachi Kaoutar
Techer Luc

STI, 4A

Année Universitaire 2020 - 2021

version : 25 mars 2021

Encadrant : Toinard Christian

Table des matières

1 Contexte

Les processeurs, dont l'évolution a été caractérisée par une augmentation continue de la fréquence de fonctionnement, suivent depuis quelques années une nouvelle voie, celle du multi-coeur. Les limites acceptables étant atteintes, la multiplication des processeurs dans un même système offre une autre possibilité d'augmentation de la puissance de calcul.
Un système d'exploitation est principalement composé d'un noyau. Celui-ci est une couche d'abstraction entre le matériel (processeur, mémoire) et le logiciel (application, *user space*), et permet leur communication. Il peut être monolithique, c'est-à-dire un programme qui est tel qu'il est et ne peut pas être modifié, pas d'ajout de fonctionnalités possible sans le recompiler. Il peut également être modulaire, qui signifie que l'on peut lui ajouter des programmes qui étendent ses fonctionnalités, et aussi les supprimer.
d'autre part le microkernel est un logiciel ou un code qui contient le minimum requis de fonctions, de données et de fonctionnalités pour implémenter un système d'exploitation. Il fournit un nombre minimal de mécanismes, ce qui est assez bon pour exécuter les fonctions les plus élémentaires d'un système d'exploitation. Il permet à d'autres parties du système d'exploitation d'être implémentées car il n'impose pas beaucoup de politiques.
Un Microkernel est la partie la plus importante pour une implémentation correcte d'un système d'exploitation, on peut voir dans le diagramme ci-dessous, que Microkernel accomplit des opérations de base comme la mémoire, les mécanismes de planification de processus et la communication inter-processus.

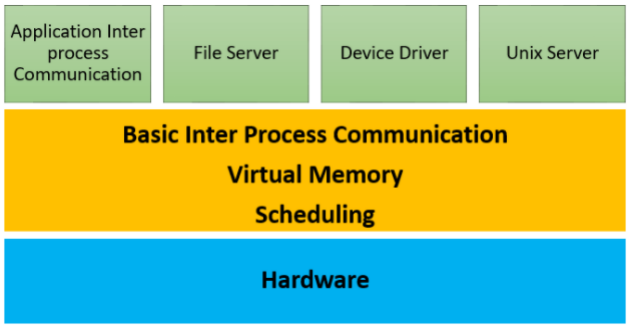

FIGURE 1 – Microkernel Based Operating System

Distributed computing est une technologie beaucoup plus large qui existe depuis plus de trois décennies de maintenant. En termes simples, il est un calcul sur des ordinateurs autonomes distribués qui ne communiquent que sur un réseau, ces systèmes sont généralement traités différemment des systèmes informatiques parallèles ou des systèmes à mémoire partagée, où plusieurs ordinateurs partagent un pool de mémoire commun utilisé pour la communication entre les processeurs. Les systèmes de mémoire distribuée utilisent plusieurs ordinateurs pour résoudre un problème commun, le calcul étant réparti entre les ordinateurs connectés (nœuds) et utilisant la transmission de messages pour communiquer entre les nœuds. Par exemple, le calcul en grille est une forme de calcul distribué où les nœuds peuvent appartenir à différents domaines administratifs. Un autre exemple est la solution de virtualisation du stockage en réseau qui utilise le calcul distribué entre des serveurs de données et de métadonnées.

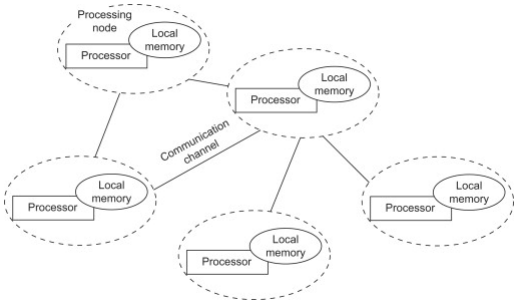

FIGURE 2 – A distributed computing system

Voici une première section. Vous pouvez faire des citations en utilisant la commande `\cite`. Par exemple [**?**].

Les informations de publication concernant le papier cité sont à placer dans un fichier `.bib`. Dans ce template, il s'agit du fichier `bibliographie.bib`.

Ces informations peuvent être obtenues sur le web, notamment ici :

https://scholar.google.fr/

Pour ce faire :

1. chercher le nom de l'article,
2. cliquez sur les guillemets,
3. puis sur BibTeX,
4. copier l'intégralité du texte dans le fichier `bibliographie.bib`,
5. modifier la clé,
6. dans votre fichier `rapport.tex`, utilisez cette clé pour citer le papier.

Exemple : New directions in Cryptography, de Diffie et Hellman

1. La recherche : https://scholar.google.fr/scholar?hl=fr&as_sdt=0%2C5&q=new+directions+in+cryptography&btnG=&oq=New+directions+in+cryptography
2. L'entrée BibTeX : https://scholar.googleusercontent.com/scholar.bib?q=info:zhumlNGssTEJ:scholar.google.com/&output=citation&scisig=AAGBfm0AAAAAXJH2ZyW scisf=4&ct=citation&cd=-1&hl=fr&scfhb=1
3. Ici, la clé par défaut est `diffie1976new`, que je modifie en `DH76`.
4. Pour citer ce papier, dans `rapport.tex`, j'utilise la commande `\cite{DH76}`

Vous pouvez aussi mettre des références en URL en pied de page [1] (mais c'est moins propre que `\cite`).

Attention, les compilations LaTeXet BibTeX peuvent être... "capricieuses". Je vous recommande de suivre cet ordre :

1. Il suffit d'utiliser la commande `\footnote` et d'inclure votre URL à l'aide de `\footnote` : https://www.latex-project.org/ ou de `\href` : même lien.

1. Compilation LaTeX : `pdflatex -synctex=1 -shell-escape -interaction=nonstopmode rapport.tex`
2. Compilation BibTeX : `bibtex rapport.aux`
3. Compilation LaTeX
4. Compilation LaTeX

Ou plus simplement, utilisez le `Makefile` fourni.

2 Problématique

Dans les années 1990, tous les systèmes d'exploitation étaient basés sur des noyaux monolithiques ce qui limitait grandement les possibilités de développement de systèmes en rendant compliqué l'intégration de tels noyaux sur des architectures matérielles complexes. Les développeurs avaient alors besoin d'un système d'exploitation fiable, sécurisé et modulable afin de pouvoir l'exploiter au mieux dans leurs applications.
De plus, un noyau monolithique limite la compatibilité et l'optimisation vis-à-vis d'architectures variées. A l'époque, les ordinateurs hautes performances et les super-ordinateurs possédaient plusieurs processeurs tandis que les ordinateurs grand-public n'en possaidaient qu'un. L'objectif de cette recherche était donc de créer un noyau modulaire qui pouvait être facilement adapté à différentes architectures plus ou moins complexes.

2.1 Une première sous-partie

Un premier paragraphe...

Un second...

2.2 Une seconde sous-partie

Inclusion d'images/screenshot ← on peut donner des titres aux paragraphes :)
Dans ce paragraphe, on va inclure une petite image (centrée) :

FIGURE 3 – Avec une légende :)

Et plus loin on peut même faire (et simplement) référence à la Figure 3 page 4.
Les formules de maths sont entre $ comme ceci $\exp^{i\pi} +1 = 0$ ou encore entre $$ pour les centrer :

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6}.$$

On peut également utiliser un environnement dédié :

$$\mathbb{Z}/p\mathbb{Z} = \{0, 1, \ldots, p-1\} \tag{1}$$

Et même aligner les équations simplement et proprement avec un autre environnement :

$$
\begin{aligned}
t &= a + b + c && (2)\\
&= d + e && (3)\\
&= z^{x \times y} && (4)\\
&= \left(\frac{\delta + \omega}{\tau} \right) && (5)
\end{aligned}
$$

Et faire références à ces équations (1) et (4).

3 Apports scientifiques principaux de l'article

3.1 Le concept des micro-noyaux

Dans cet article, le concept des micro-noyaux est introduit. Il s'agit d'une solution qui consiste à réduire au plus possible le noyau (aussi appelé Nucleus) afin que celui-ci ne puisse effectuer que des tâches élémentaires. Les autres fonctionnalités seront quant à elles effectuées par des serveurs modulaires. Ces serveurs sont en réalité des sous-systèmes qui pourront échanger des informations entre-eux en utilisant des Communications Inter-Processus (IPC). Ces différents sous-systèmes pourront alors être répartis sur un seul ou plusieurs processeurs ou machines. Afin de faciliter les communications entre les différents serveurs, l'article propose aussi la mise en place de Remote Procedure Call (RPC) qui est un protocole qui permet d'exécuter des commandes sur un serveur à distance.
La mise en place de micro-noyaux permettrait de faciliter grandement la répartition et l'isolations de différentes parties du systèmes sur des processeurs ou machines différentes tout en restant compatible avec des architectures plus simples. Ceci permettrait aussi aux développeur de pouvoir créer, tester et implémenter de nouvelles fonctionnalités beaucoup plus simplement.

3.2 Présentation de Chorus

Chorus est un système d'exploitation novateur pour l'époque créé par le laboratoire de recherche INRIA (France) puis développé d'avantage et commercialisé par CHORUS Systems. Il s'agit de l'exemple d'implémentation des micro-noyaux décrite dans l'article. A l'époque, c'était l'une des rare distribution à explorer le domaine des micro-noyaux avec quelques autres exemples notoires tels que Amoeba, Topaz et System-V. Chorus fut une distribution novatrice dont le développement a malheureusement été abandonné depuis. Cependant, Chorus a constitué une base solide pour le développement des micro-noyaux. L'architecture de Chorus est constituée, comme décrit précédemment, d'un micro-noyau qui peut effectuer des fonctionnalités élémentaires et de serveurs qui viennent compléter ce Nucleus.

Chorus-v3 était une référence en la matière car il s'agissait de la 4sup è*me* génération de la distribution. De plus elle faisait usage de contributions de la part de System-V notament dans ce qui est des communications inter-processus et des RPC, de Mach pour ce qui est de la gestion de la mémoire virtuelle partagée et des threads, Amoeba pour la gestion d'addresse et les capacités de binding. Chorus a donc essayé de prendre le meilleur des différentes distributions existantes afin de créé une référence en terme de système d'exploitation utilisant les micro-noyaux.

Le Nucleus de Chorus est un noyau temps réel qui implémente le calcul distribué et les communications de bas niveau. Il est lui-même composé de 4 composants différents :

— Un puissant ordonanceur permettant la gestion des différents processeurs et threads afin de pouvoir exécuter les tâches de manière optimisée et distribuée ;
— Un gestionnaire de mémoire distribuée qui supporte toutes les architectures mémoires ;
— Un superviseur d'évènement de bas niveau qui permet de distribuer les intéruptions système, trapes et exeptions venant de l'extérieur sur des ports ou routines définis dynamiquement ;
— Un gestionnaire de communication inter-processus qui fournit d'excélentes performances de communications entre les différents services.

Les sous-systèmes utilisés par Chorus sont en quelques sortes des morceaux de systèmes d'exploitation conventionnels qui agissent par dessus le Nucleus. Ils permettent la gestion de ressources physiques et logiques telles que la gestions des fichiers, des gestionnaires de fichiers, des appareils ainsi que des communication de plus haut niveau. L'architecture est telle qu'un processus ne peut pas voir un sous-système comme un fragment du noyau mais bien comme un système d'exploitation standart. Ceci est dû aux interfaces proposées par les sous-systèmes qui doivent être rigoureusement les mêmes que celles offertes par un système classique. Chaque sous-système possède donc son interface qui interagit avec ses composants et le Nucleus. Cette architecture permet, en plus de faciliter la distribution du système, de grandement simplifier le développement système en faisant abstraction des fonctionnalités de très bas niveau étant donné qu'elles sont déjà implémentées dans le Nucleus.

4 Impacts de l'article

5 Analyse critique du travail proposé

Conclusion

Ce magnifique projet nous a permis – outre de nous familiariser avec LaTeX – de ...

Annexes

A Algorithme qui fait quelque chose

```c
#include <stdio.h>

int main()
{
        printf("Hello World !\n);
        return 0;
}
```

B Une autre annexe