La virtualisation lourde

Pré-requis lexical

Afin de correctement comprendre l'ensemble de ce texte, voici une brève définition des différents termes techniques qui y seront utilisés :

- **Noyau**: le noyau est le coeur du système d'exploitation. Il gère les ressources matérielles de la machine sur laquelle il est installé et assure la communication entre les différentes programmes et le matériel.

- **Machine hôte**: la machine hôte est la machine physique sur laquelle se trouve les différentes machines virtuelles. Son système d'exploitation sera désigné comme système hôte.

- **Machine virtuelle**: Une machine virtuelle n'existe que de manière logique grâce à des programmes comme un émulateur ou un hyperviseur. Son système d'exploitation sera désigné comme système invité.

- **Hyperviseur**: Dans le cadre d'une virtualisation lourde, un hyperviseur est un programme qui à la responsabilité de gérer les machines virtuelles et de réceptionner les appels systèmes d'un système d'exploitation invité, afin de les modifier pour qu'ils soient compréhensibles par le système hôte et

inversement, ou bien directement par le matériel dans le cadre de la paravirtualisation.

- **Orchestration**: de manière général, l'orchestration est un mécanisme logique qui alloue la puissance de calcul à chaque processus en nécessitant, de manière équilibré et en fonction des besoins et de la priorité du processus. Dans le cadre d'une virtualisation lourde, l'hyperviseur fait office d'orchestrateur pour les différentes machines virtuelles dont il a la charge. Il fait en sorte d'équilibrer l'attribution des ressources du système hôte à chacune d'entre elles.
  Dans le cadre de la paravirtualisation (notamment réseau), où les machines virtuelles, le(s) hyperviseur(s) et l'espace de stockage se trouvent sur des machines physique différentes, l'orchestrateur est un programme qui va faire office de lien entre ces différents éléments afin d'éviter tout conflit et de justement distribuer les ressources matérielles à ces différents services.

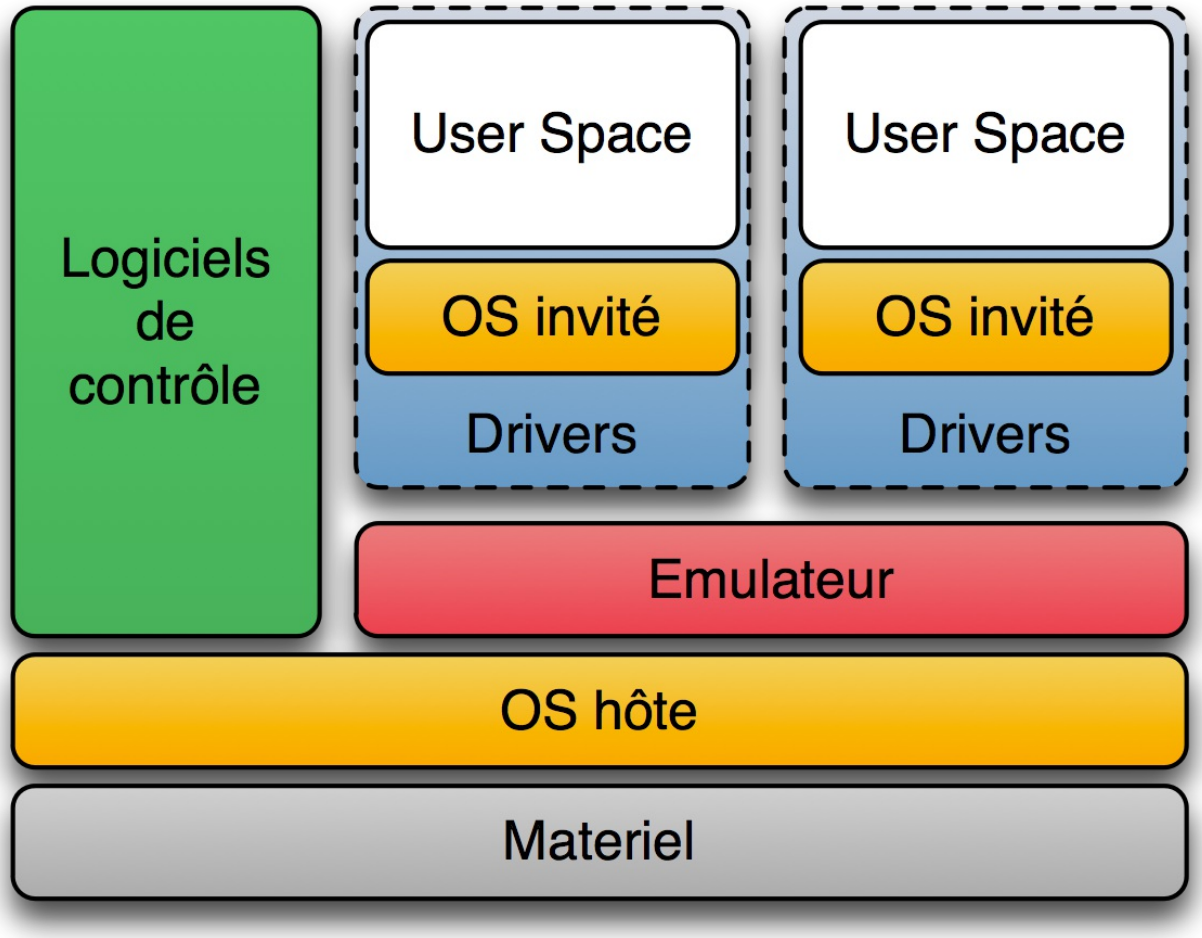
- **Anneaux de protection**: Les anneaux de protection sont des niveaux de privilège imposés par l'architecture d'un processeur. Généralement, l'anneau possédant le plus de privilèges se situe au niveau 0. C'est ce dernier qui se charge de l'accès au matériel.
  A l'inverse, l'anneau étant au plus haut niveau englobe tous les programmes.

Rappel des différents types de virtualisation

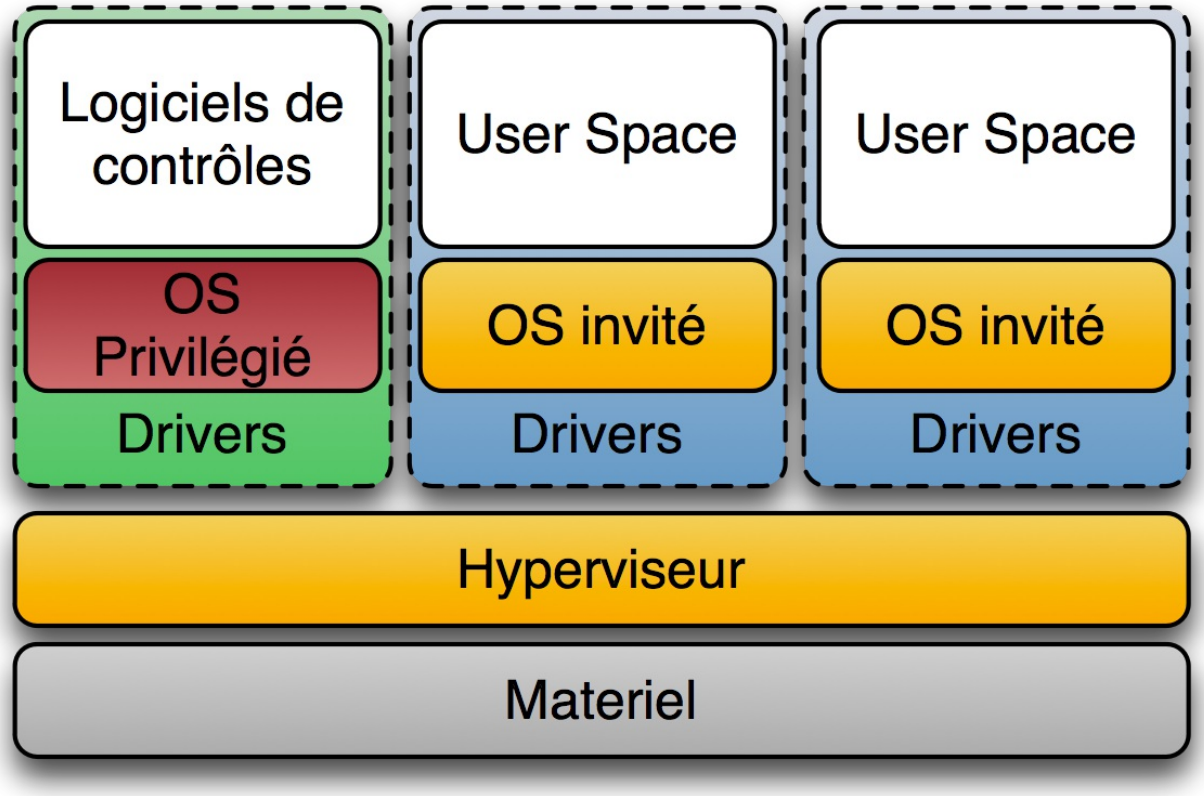
La virtualisation, du point de vue machine, est le fait de faire fonctionner un ou plusieurs systèmes d'exploitation sur la même machine.

Ces machines peuvent être virtualisées à différents niveaux :

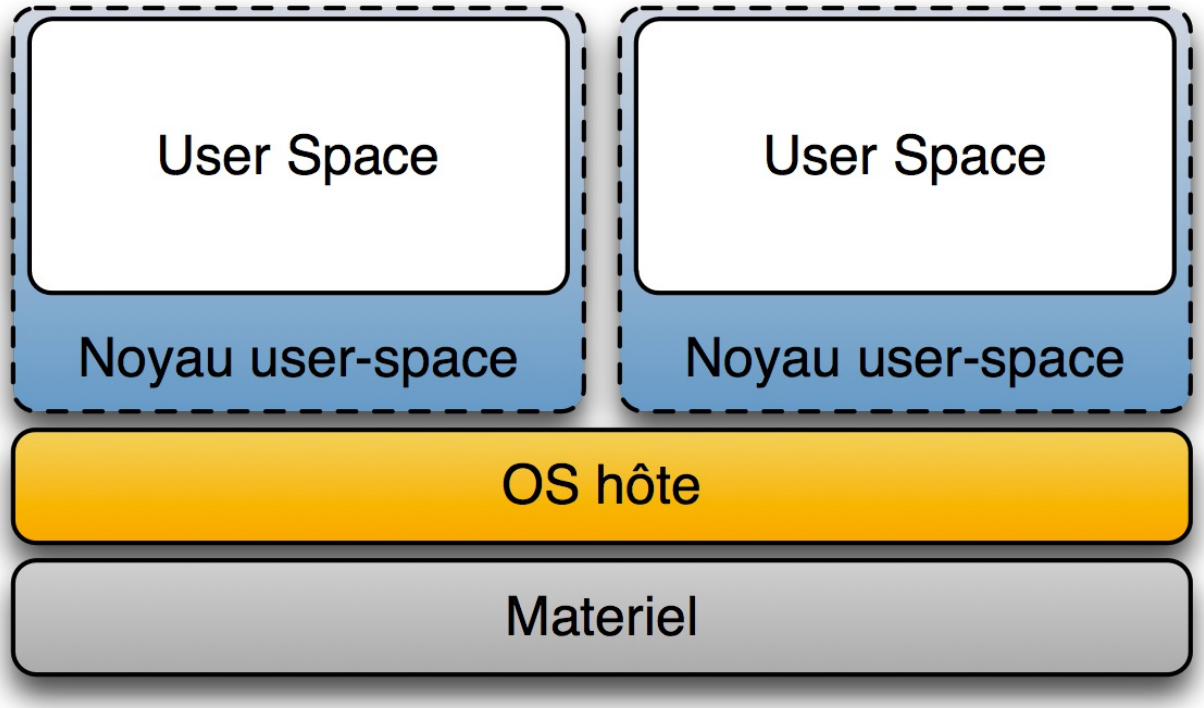
- virtualisation d'un ordinateur complet avec tout son matériel (processeur, RAM, espace de stockage, etc…). Cette solution est la plus compatible si l'on souhaite utiliser un système d'exploitation qui n'a pas été prévu spécifiquement pour cette situation, car faisant abstraction du matériel physique de la machine hôte. Cependant, elle est aussi la plus lente et gourmande en terme de ressources de par l'imitation d'un matériel quasi-complet via un hyperviseur de type 2, d'où sa dénomination de virtualisation "lourde".



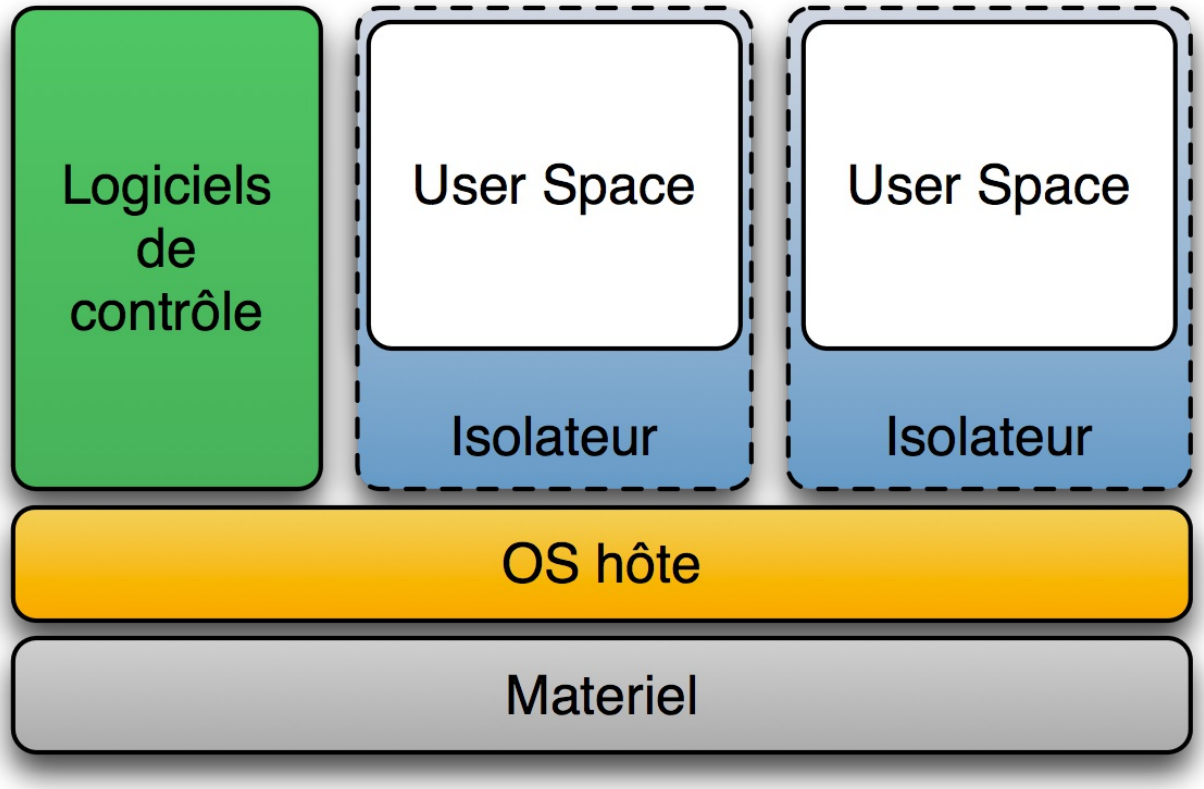
- virtualisation d’un système d’exploitation prévu à cet effet qui utilise directement les ressources matérielles offertes par la machine hôte à l’aide d’un hyperviseur de type 1. Cette solution est la plus performante car les systèmes d’exploitation sont optimisés pour ça. En contrepartie, c’est aussi la plus contraignante et onéreuse, particulièrement si l’on utilise des logiciels propriétaires payants. On qualifie cette solution de “paravirtualisation”.



- virtualisation d’un environnement système (un noyeau) directement dans l’espace de l’utilisateur comme n’importe quel autre logiciel. Ce nouvel environnement dispose ensuite de son propre espace utilisateur et gère ses applications comme s’il était le noyau légitime de la machine. Cette solution est la moins performante. Les causes sont multiples : la présence de deux noyaux sur la même machine, le manque d’isolation entre ces derniers et par conséquent la dépendance du noyau émulé par rapport au noyau hôte.



- virtualisation d’un ensemble d’applications par un isolateur. Bien qu’il ne s’agisse pas de “virtualisation de système” à proprement parler, cette solution propre à GNU/Linux permet de faire fonctionner un lots d’applications, voir même plusieurs instances de la même application, sans avoir à se préoccuper du matériel de la machine hôte. Ceci est très performant, mais l’isolation n’étant pas complète, cette solution est totalement dépendante du noyau hôte.



Ainsi dans le cadre de la virtualisation lourde, une machine vituelle est un ordinateur quasi-complet disposant d'un processeur, de mémoire vive, un ou plusieurs disques durs, une ou plusieurs interfaces réseaux et plus ou moins de périphériques externes (lecteur de disquette, CD-ROM, contrôleur USB, etc…). Cet "ordinateur" est géré par un hyperviseur de type 2 qui se charge de faire le lien entre la machine virtuelle et la machine hôte. Lors de la création de la machine virtuelle, l'utilisateur se retrouve tout simplement avec une machine dont le disque dur est vide. Il lui incombe d'installer le système d'exploitation de son choix, comme s'il venait réellemment de se procurer un ordinateur non-formaté.

Il est à noter que le système d'exploitation de la machine virtuelle doit être prévu pour la même architecture processeur que celle de l'hôte,

car les logiciels de virtualisation se contentent de donner au processeur les instructions basiques émanant du système invité. En effet, si le système invité n'est pas prévu pour l'architecture du processeur de la machine hôte, il en résultera des crashs puisque le processeur recevra des instructions qui ne font pas partie de son propre jeu d'instructions et ne saura pas comment les traiter. La solution à ce problème s'appelle l'émulation qui elle va réinterpreté les instructions du système invité pour qu'elles soient compréhensibles par le processeur de la machine hôte.

Les hyperviseurs

Comme mentionné précédemment, il existe deux types d'hyperviseurs:

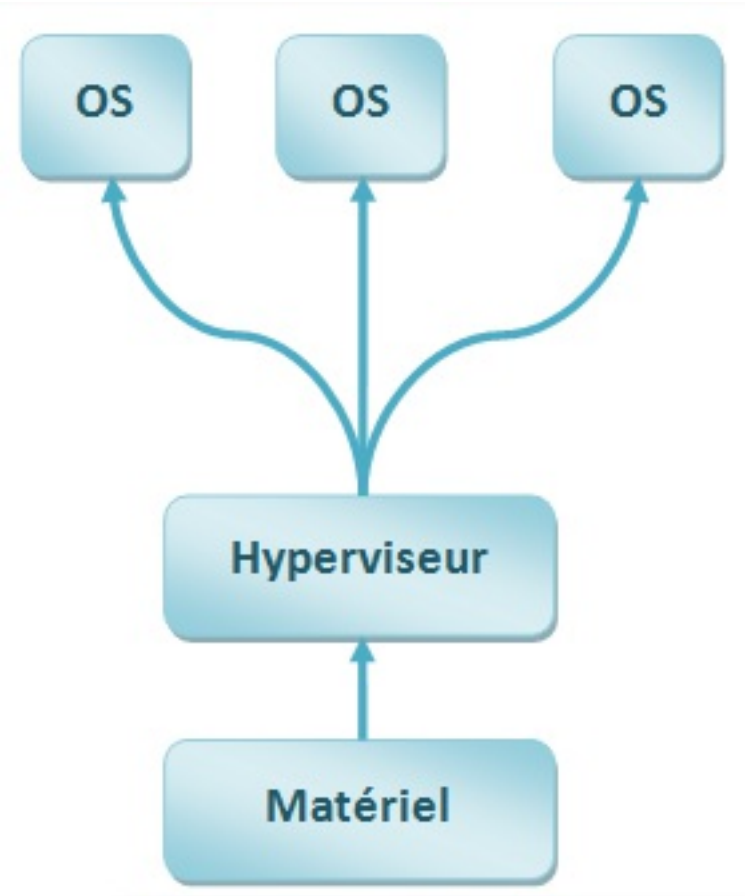
- L'hyperviseur type 1 (natif)
- L'hyperviseur type 2 (logiciel)

L'hyperviseur est un plate-forme applicative permettant de faire cohabiter plusieurs systèmes d'exploitation sur une même machine (virtuelle ou physique).

Le principe d'hypervision est aussi présent dans le domaine des SI, mais c'est un concept différent.

Ici, l'hyperviseur n'aura que pour but d'accueillir ces machines et d'allouer les ressources nécessaires au fonctionnement de celles-ci (dès le démarrage de la machine hôte pour le type 1 ou bien du logiciel de virtualisation pour le type 2).

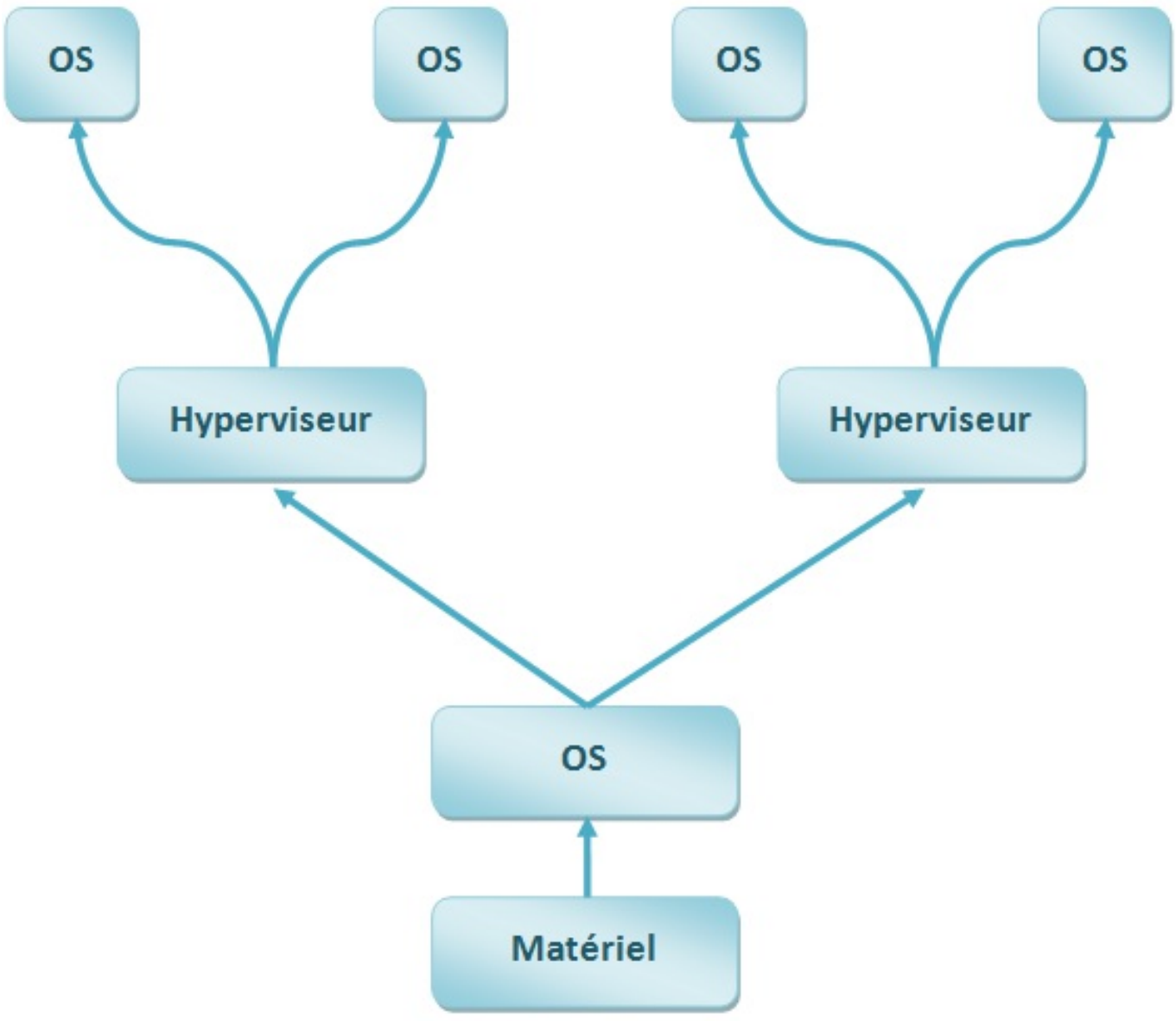
L’hyperviseur de type 1 est directement installé sur le matériel de la machine hôte, il contrôle donc les systèmes d’exploitation qui y sont installé. Il est considéré par ces dernier comme un noyau léger, optimisé pour virtualiser des machines.
Sur les processeurs prévus pour la virtualisation (instructions de virtualisation matérielle), les anneaux de protection n’ont plus à être émulés, ce qui accélère le fonctionnement de l’hyperviseur.



L’avantage de ce type d’hyperviseur est que la majeure partie des ressources disponibles sont allouables aux différentes machines virtualisées. Cela a pour conséquence de pouvoir faire fonctionner de gros serveurs très gourmands en ressources sur le même hôte, à condition bien sûr que ce dernier possède à lui seul la puissance nécessaire pour faire tourner ces machines.

Comme exemples d’hyperviseur de type 1, nous pouvons citer Citrix XenServer ou Microsoft Hyper-V.

Pour l’hyperviseur de type 2, il s’agit d’un logiciel de virtualisation. Il n’est donc plus en lien direct avec le matériel. Il s’installe et s’exécute au sein du système hôte (déjà installé) et ne le contrôle donc pas. Il se situe au 3ème niveau (au dessus du matériel).
Via une interface souvent plus abordable, les machines sont moins performantes, dû au fait que le système hôte en consomme déjà une partie pour fonctionner.



Son principal avantage réside dans le fait que les systèmes invités n’ont pas conscience d’être virtualisés, il donc n’est pas nécessaire qu’ils soient adaptés à la virtualisation, ce qui permet de faire tourner de vieux systèmes d’exploitation selon les besoins.
Il est nécessaire de préciser qu’avec ce type d’hyperviseur, on peut installer et exécuter autant d’hyperviseurs que l’on souhaite (selon les ressources disponibles). Ici, nous citerons VMWare Workstation, VirtualPC ou le plus connu VirtualBox.

Evolution technologique de la virtualisation

Un peu d’histoire

Le concept de virtualisation remonte au années 60 où fut mené une grande part des travaux la concernant au centre scientifique de Cambridge d’IBM avec le MIT. Ces recherches donnèrent un système expérimental qui devint une gamme de produit vendue par IBM à partir de 1964, alors désignée comme Hyperviseur, aujourd’hui appellée “mainframe” : la série System/360 puis 370. Le principe de cette gamme n’était autre que de couvrir tout les besoins informatiques, scientifiques et de gestion de l’époque, tout en étant compatible avec l’ensemble des autres modèles de la gamme. Cette virtualisation était d’ordre mécanique, c’est-à-dire que l’ensemble des machines de cette gamme utilisait la même architecture, créant sans le savoir les prémices de ce qui serait le compatible-PC. La génération suivante de supercalculateur, la Série Z devenue System Z en avril

2006, prolonge cette philosophie en rendant aisé le changement d'une machine à une autre au sein de la série. De plus, cette série est justement majoritairement dédiée à la paravirtualisation.

Dans les années 80, sur certains micro-ordinateurs de l'époque furent expérimentées des techniques de virtualisation, de manière logicielle ou matérielle par l'ajout de composant. On pourra notamment citer l'exemple de l'ADAM pour la virtualisation matérielle. Deux systèmes étaient disponibles :

- sous forme de pack d'extension pour la console CBS Colecovision, la transformant alors en une suite bureautique pour les standards de l'époque
- soit directement avec la console intégrée sous forme de circuit imprimé

L'ADAM était composé d'une imprimante à marguerite, un clavier professionnel, un lecteur de cassettes et trois programmes (un traitement de texte en ROM, un compilateur BASIC et le jeu Super Buck Rogers sur cassette).

Mais la machine ayant véritablement tiré son épingle du jeu dans ce domaine à ce moment n'est autre que l'Amiga, vendue par Commodore International de 1985 à 1994, sur laquelle il était possible de lancer en multitâche d'autres systèmes d'exploitation comme Windows, GNU/Linux ou encore Macintosh à partir d'AmigaOs, son système d'exploitation natif.

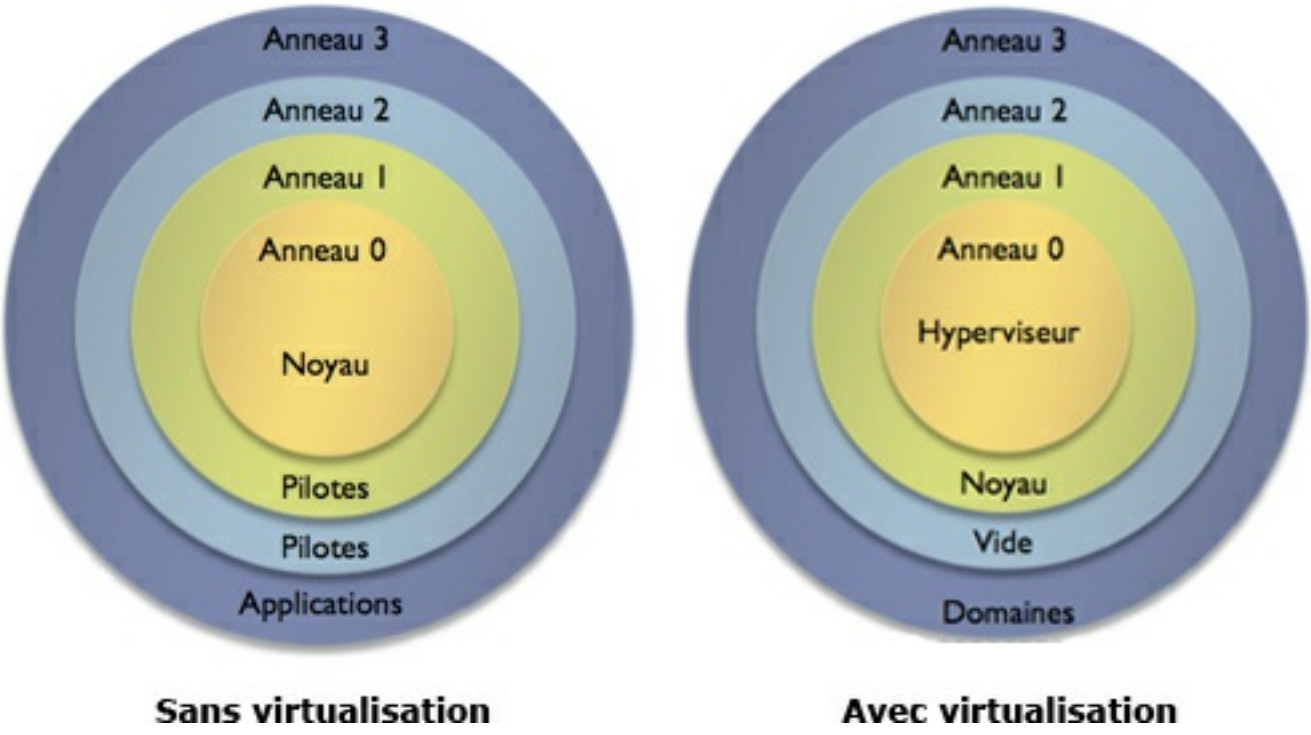
A partir de la seconde moitié des années 90, des passionnés et nostalgiques se lancent dans des projets d'émulation de machines de la décennie précédente, comme le C64 ou la NES. Enfin, à l'aube des années 2000, la société VMware lance et popularise un système de virtualisation lourde propriétaire et payant pour les architectures de type x86, qui était le standard des compatible-PC. Devant le succès de ce procédé, d'autres projets du même acabit voit le jour comme le logiciel libre VirtualBox maintenant détenu par Oracle, ou encore VirtualPC, spécialisé pour une utilisation des différentes versions de Windows et créé par Connectix, racheté en octobre 2003 par Microsoft.

Souci technique

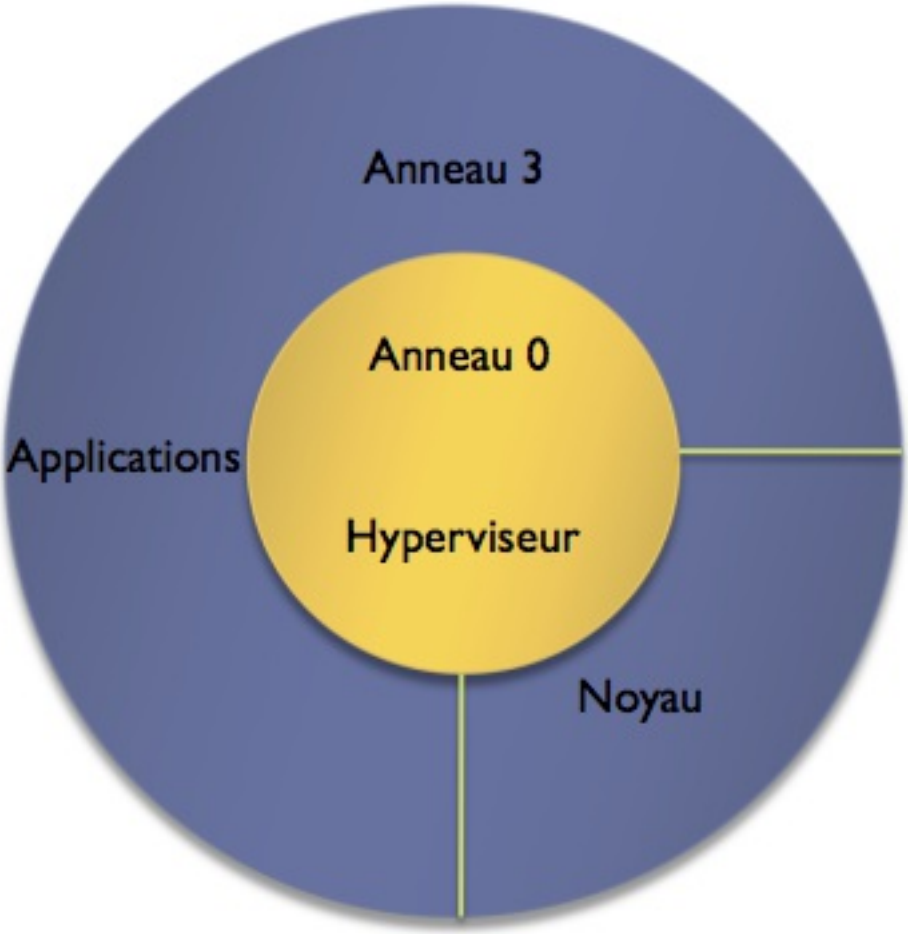
Historiquement, l'architecture standard des compatible-PC est celle du x86, basée sur le jeu d'instructions du processeur 8086 d'Intel créé en 1978. Au niveau des systèmes d'exploitation 32 bits, cette architecture met à disposition des anneaux de protection des données en mémoire (au nombre de quatre). Le noyau 0 étant généralement utilisé par le noyau du système afin d'accéder au matériel et le 3 pour les programmes. L'objectif est de cloisonner les données des différents programmes afin que jamais ils n'accèdent directement à celles se trouvant dans d'autres anneaux. C'est justement le rôle du système d'exploitation de faire office de pont entre les anneaux lorsque cela est nécessaire.

Remarquant que les anneaux 1 et 2 était rarement utilisés, les

principaux fabricants de processeurs, Intel et AMD, les ont simplement supprimé lors de la mise au point de l'architecture x64.



Cela posera un problème pour les logiciels de virtualisation qui plaçaient justement leur hyperviseur dans l'un de ces deux noyaux par mesure de sécurité. Les éditeurs de ces logiciels trouvèrent comme solution de placer l'hyperviseur dans l'anneau 0 et le noyeau du sytème dans le même anneau que les applications (et inversement dans le cadre d'une virtualisation lourde), tout en faisant en sorte de le cloisonner au maximum pour que les applications n'y aient pas directement accès.

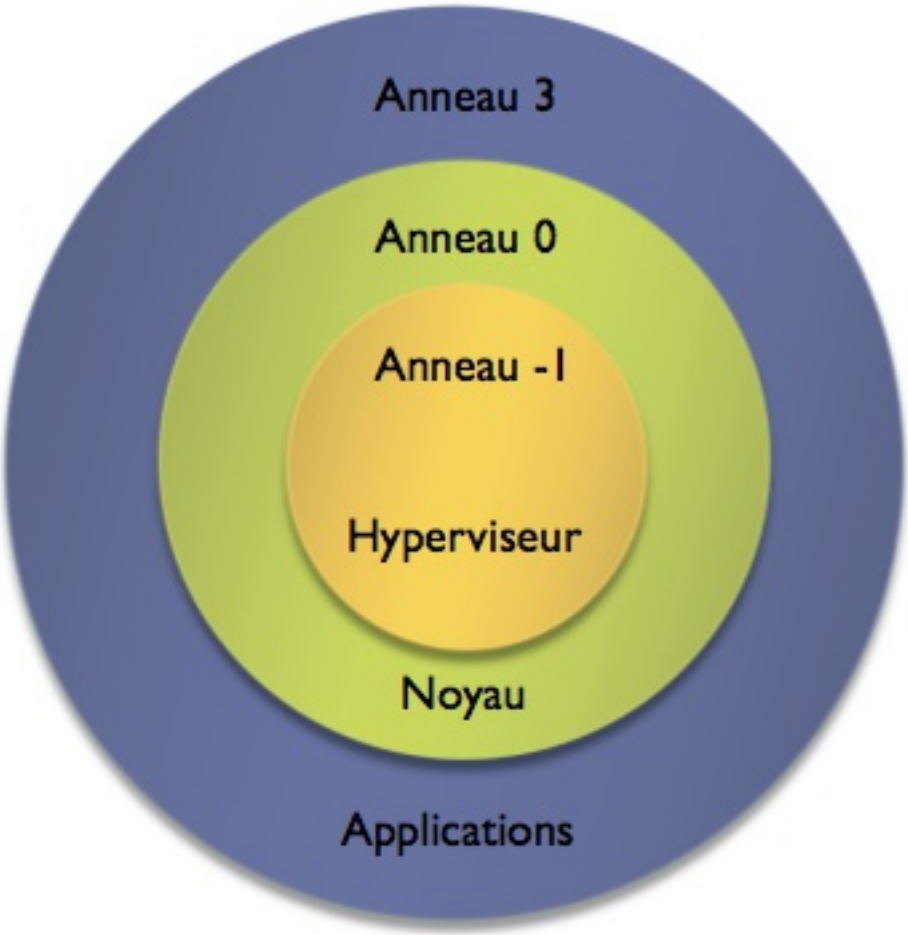


Face à l'importance de plus en plus croissante de la virtualisation, notamment dans le domaine professionnel, Intel et AMD modifièrent leurs processeurs pour étendre leur jeu d'instructions en y ajoutant des fonctionnalités de virtualisation matériellement assistée, parmi celle-ci on trouve:

- l'ajout d'un anneau -1 pour y placer l'hyperviseur ou le noyau afin qu'il soit à nouveau correctement isolé seul dans son anneau.

- l'ajout pour l'hyperviseur d'un accès direct aux ressources matérielles afin de mieux les redistribuer en fonction des besoins de chaque machine virtuelle

- la permission aux machines virtuelles de ne plus êtres totalement dépendantes du noyau du système hôte puisque l'hyperviseur dispose d'un accès direct au processeur



Exemple par la pratique

Dans cet exemple, nous allons créer et configurer en ligne de commande une machine virtuelle (virtualisation lourde) à l'aide du logiciel VirtualBox d'Oracle via son utilitaire intégré VBoxManage. Afin de prouver que le système invité n'est pas dépendant du système hôte, nous y installerons Windows XP 32bits.

La procédure d'installation d'un programme étant propre au système

d’exploitation que vous utilisez ainsi qu’à vos préférences pour telle ou telle procédure, nous n’aborderons pas cette étape ici.

Nous précisons par ailleurs que cet exemple est effectué sous Ubuntu 16.04 LTS 64bits avec la version 5.0.24_Ubuntu r108355 de VirtualBox sur une machine possédant une architecture x64 avec un Intel Core I5 prenant en charge la virtualisation matériellement assitée ainsi que 8Go de mémoire vive.

Tout d’abord, il faut créer le fichier de configuration de la machine virtuelle que nous nommerons winXP et que nous attacherons directement via à console de VirtualBox avec `--register`:

```
VBoxManage createvm --name winXP --register
```

Ce fichier se trouvera par défaut dans le dossier `VirtualBox VMs/winXP` dans votre répertoire personnel.

Vous pouvez créer ce fichier sans l’attacher à cette console mais il faudra alors taper la commande suivante pour l’enregistrer:

```
VBoxManage registervm '/votre/home/VirtualBox VMs/winXP/winXP.vbox'
```

Nous allons ensuite créer le fichier qui représente le disque dur virtuel de la machine. Ceci n’étant qu’un exemple, nous lui attribuerons 5GB. Ce fichier se trouvera dans le répertoire courant au moment de

l’exécution de la commande:

```
VBoxManage createhd --filename winXP --size 5000
```

Configuration en ligne de commande de notre machine virtuelle

Maintenant il faut définir le type de système d’exploitation que nous souhaitons utiliser:

```
VBoxManage modifyvm winXP --ostype WindowsXP
```

Pour une liste de l’ensemble des types de système d’exploitation pris en charge:

```
VBoxManage list ostypes
```

Nous lui allouerons 100MB de mémoire vive:

```
VBoxManage modifyvm winXP --memory 100
```

Passons au contrôleur de stockage. Ici il s’appellera IDE pour un contrôleur de stockage de type ide, avec un chipset émulé PIIX4 pour y connecter ensuite un disque dur et un lecteur DVDROM tout en étant amorçable, c’est-à-dire qu’il sera lancé par le BIOS au démarrage de la machine:

```
VBoxManage storagectl winXP --name IDE --add ide --contro
ller PIIX4 --portcount 2 --bootable on
```

Remarque: vous pouvez très bien chosir un contrôleur de type sata et préciser le nombre de port dont celui-ci dispose avec `--portcount <1-30>`. Ici nous choisissons ide car notre version de l'installeur de Windows XP ne dispose pas de pilote pour le sata et plante donc dès le démarrage.

Nous allons tout d'abord y attacher un disque dur qui sera représenté par le fichier de disque dur virtuel que nous avons créé plus tôt:

```
VBoxManage storageattach winXP --storagectl IDE --port 0
--device 0 --type hdd --medium "/chemin/absolu/winXP.vdi"
```

Puis y attacher un lecteur DVDROM avec l'image disque du CDROM d'installation de Windows XP:

```
VBoxManage storageattach winXP --storagectl IDE --port 1
--device 0 --type dvddrive --medium "/chemin/absolu/image
/disque/xppro.iso"
```

`--medium` peut prendre la valeur `emptydrive` si vous ne voulez pas y mettre d'image disque, `host:/chemin/lecteur/physique` pour que la machine virtuelle utilise un lecteur de la machine hôte ou encore `addition` pour installer les addons invités permettant le partage de

fichiers entre le système hôte et l'invité ou encore une meilleur gestion de capture des entrées entre les deux systèmes.

Nous allons ensuite configurer l'affichage et le son, notamment l'accélération 3D (si vous en disposez), 100 Mo de mémoire vidéo (Vram), le pilote audio et son codec:

```
VBoxManage modifyvm winXP --vram 100 --accelerate3d on --
audio alsa --audiocontroller ac97
```

Enfin, la machine virtuelle disposera d'une seule carte réseau de type PCnet-FAST III, qui sera relié au réseau, et ce par la méthode NAT qui est la plus simple d'utilisation:

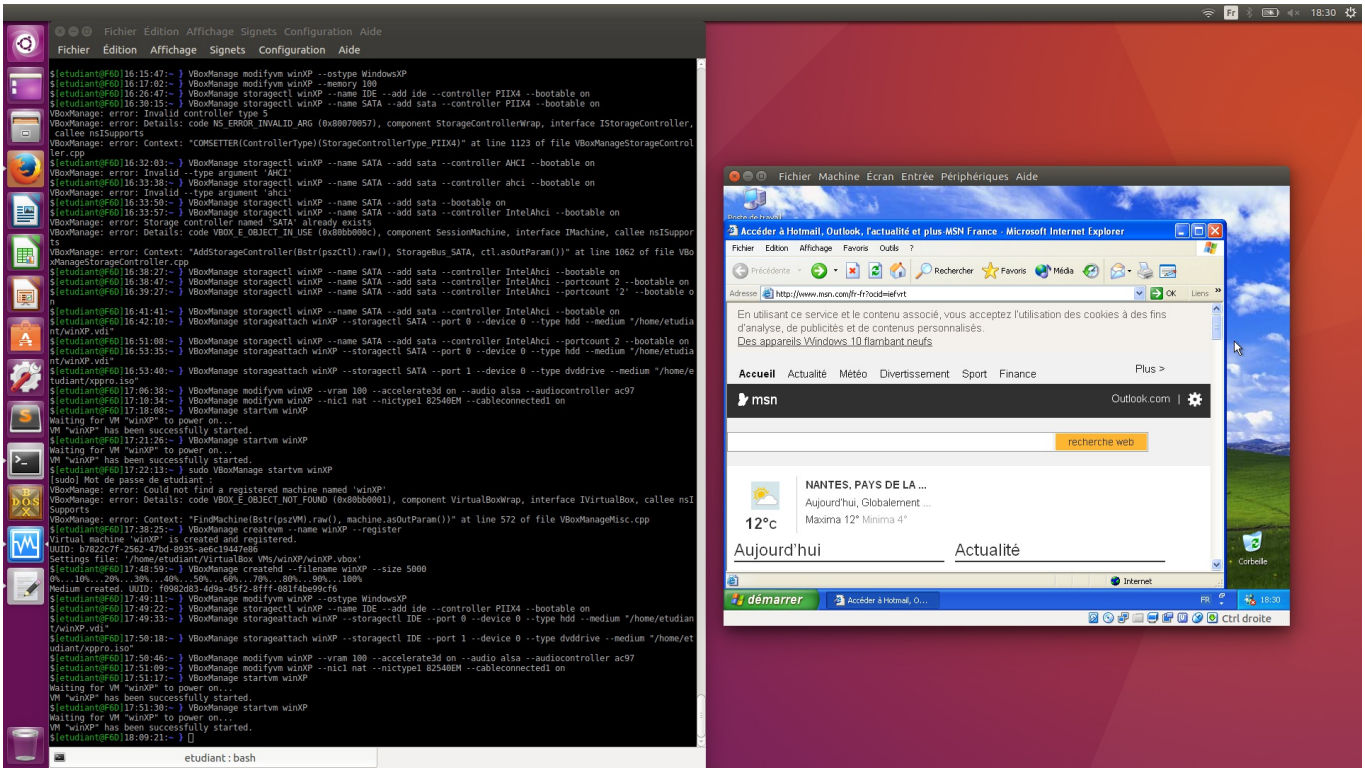
```
VBoxManage modifyvm winXP --nic1 nat --nictype1 Am79C973
--cableconnected1 on
```

La configuration physique étant terminée, nous pouvons finalement allumer la machine virtuelle:

```
VBoxManage startvm winXP
```

Le reste se passe exactement de la même manière que si vous utilisiez une vraie machine physique: installation du système d'exploitation, des programmes puis utilisation de ceux-ci. L'avantage étant que si vous compromettez le système invité, cela n'impacte nullement le système

hôte ! Il vous suffira de réinstaller le système invitée sur la machine virtuelle.



Sources

Généralités informatiques:

https://fr.wikipedia.org/wiki/Noyau_de_syst%C3%A8me_d%27exploitation

https://en.wikipedia.org/wiki/Orchestration_(computing)

https://fr.wikipedia.org/wiki/Anneau_de_protection

La virtualisation, ses différents types, et la virtualisation lourde:

https://fr.wikipedia.org/wiki/Virtualisation

https://doc.ubuntu-fr.org/virtualisation

http://www.culture-informatique.net/cest-quoi-la-virtualisation/

https://www.antoinebenkemoun.fr/2009/10/classification-des-types-de-virtualisation-mise-a-jour/

https://www.antoinebenkemoun.fr/2009/07/les-differents-types-de-virtualisation-la-virtualisation-totale/

Les hyperviseurs:

https://fr.wikipedia.org/wiki/Hyperviseur

https://www.antoinebenkemoun.fr/2009/10/de-la-differenciation-hyperviseur-type-1-type-2/

http://www.it-connect.fr/les-types-dhyperviseurs

La virtualisation matériellement assitée:

https://www.antoinebenkemoun.fr/2009/08/les-anneaux-de-protection/

https://www.antoinebenkemoun.fr/2009/08/les-anneaux-de-protection-systeme-dans-le-cas-du-64-bit/

https://www.antoinebenkemoun.fr/2009/07/la-virtualisation-materiel-assistee/

Différences entre virtualisation et émulation:

http://carvounas.net/blog/2010/05/17/virtualisation-vs-emulation/

http://www.tomshardware.fr/articles/virtualisation-Intel-AMD,2-353-4.html

L'IBM System/360, 370 et z:

https://fr.wikipedia.org/wiki/IBM_360_et_370

https://fr.wikipedia.org/wiki/ZSeries

https://fr.wikipedia.org/wiki/System_z

La Colecovision, l'ADAM et l'Amiga:

http://www.grospixels.com/site/adam.php

http://gamopat.com/article-1972191.html

https://en.wikipedia.org/wiki/Coleco_Adam

https://fr.wikipedia.org/wiki/Amiga

VirtualBox:

https://www.virtualbox.org/manual/