NSI - Première

Architecture - OS : trois concepts clés

qkzk

2020/10/12

Le systèmes d'exploitations (OS) sont des logiciels spécialisés qui servent d'intermédiaire entre le matériel et l'utilisateur.

Ils rendent l'utilisation du matériel possible et sûre.

Les grandes familles de systèmes d'exploitation modernes sont UNIX et Windows.

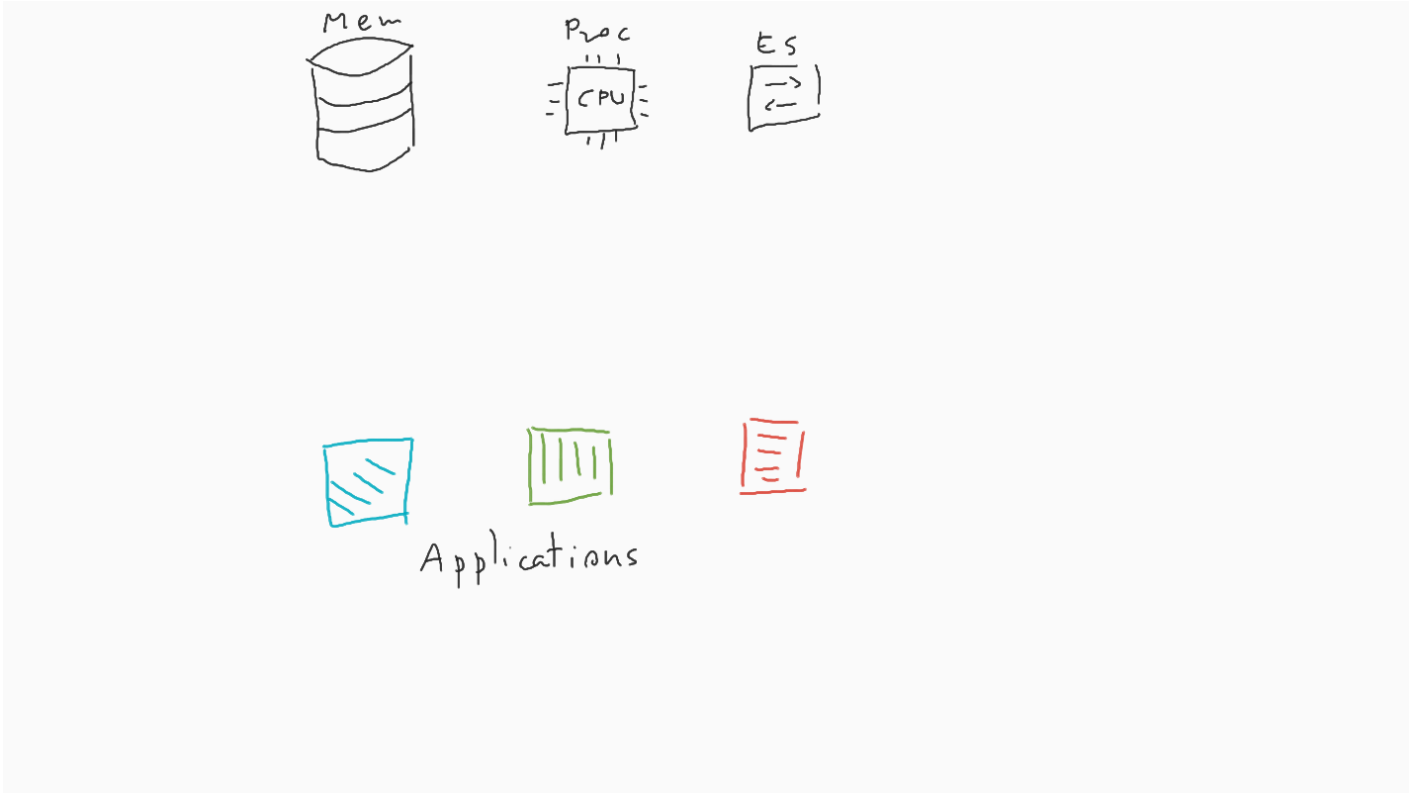
UNIX n'est pas un OS à proprement parler mais désigne tous les OS qui s'inspirent des premières versions d'UNIX : Linux, Android, iOS, MacOS etc.

La mémoire

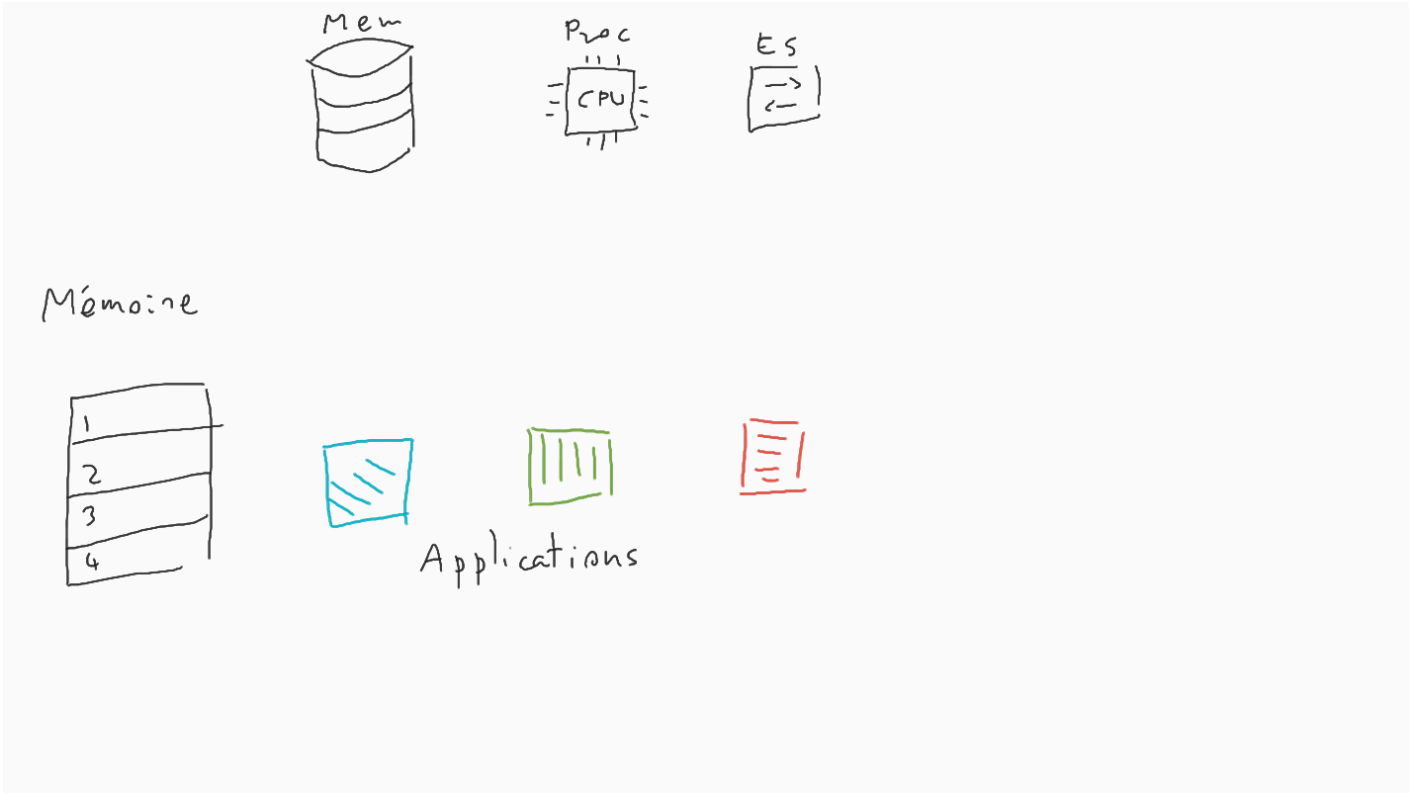
Résumons le matériel d'un ordinateur à trois grandes catégories :

- la mémoire (où sont enregistrées les informations)
- le processeur (qui calcule),
- les entrées/sorties (clavier, souris, écran, réseau, haut parleur etc.)

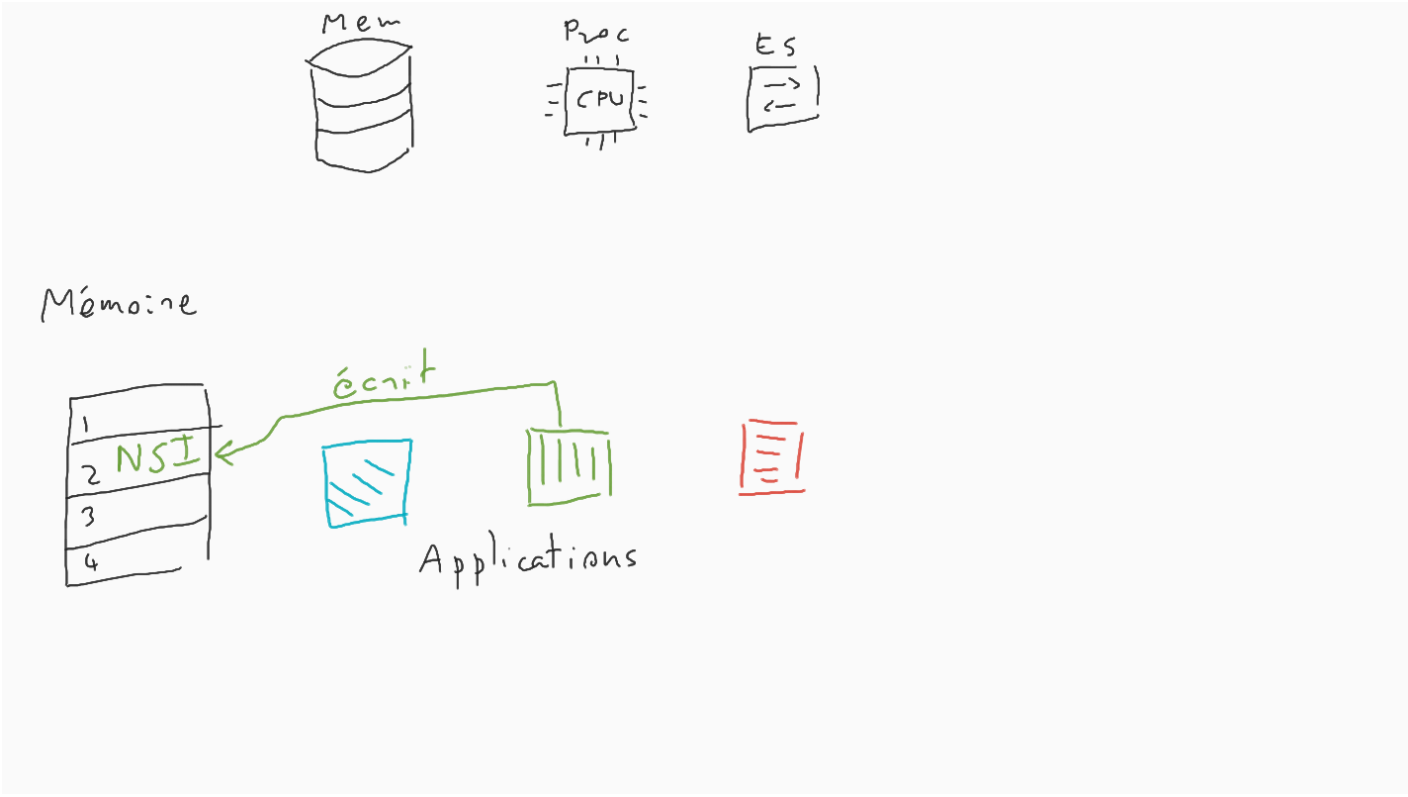
Supposons que trois applications tournent en parallèle (on en reparlera).



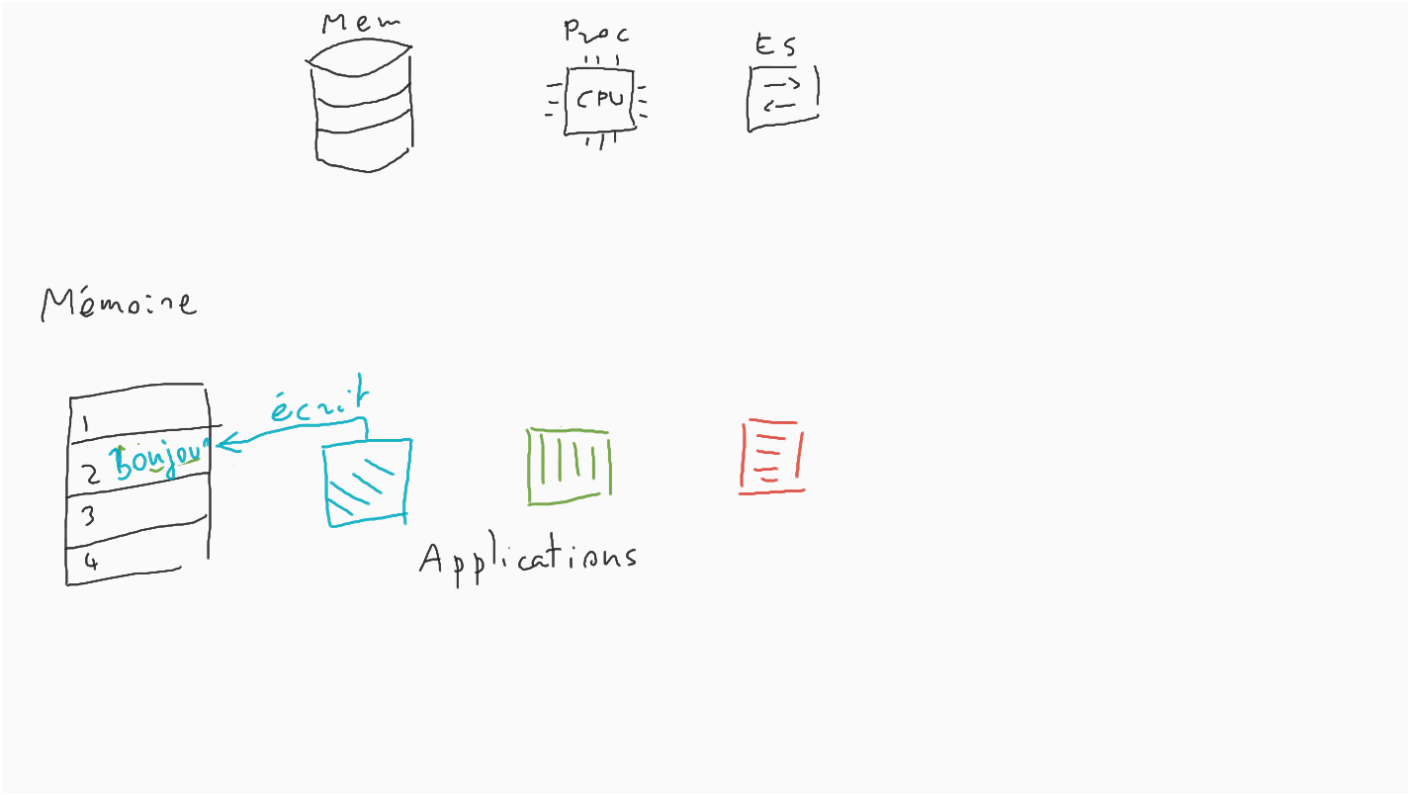
Elles disposent toutes d'un accès à la mémoire, qu'on peut voir comme un grand tableau de "mots".



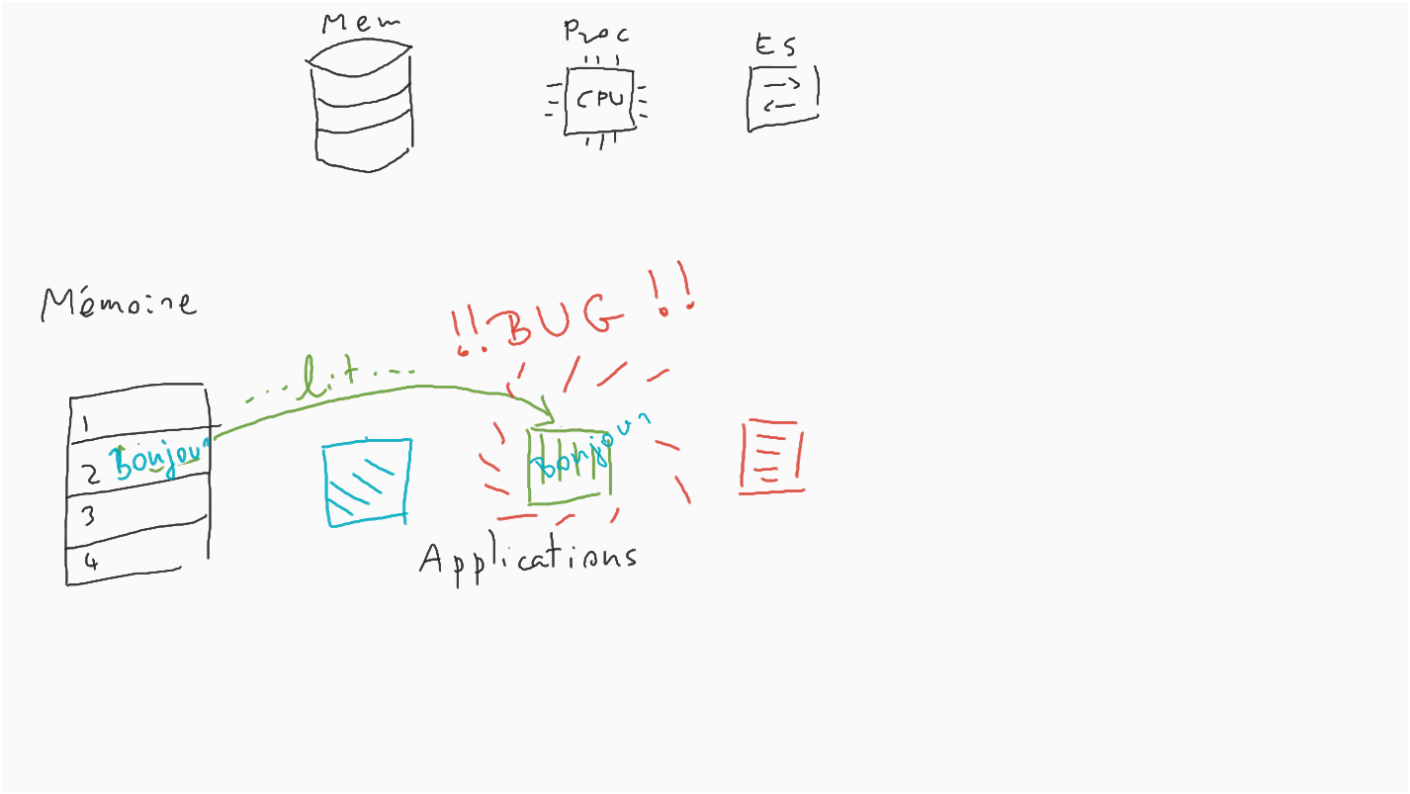
Imaginons que l'application verte écrive le mot "NSI" à l'emplacement 2 de la mémoire.



Ensuite l'application bleue écrit le mot "Bonjour" à ce même emplacement et écrase le précédent contenu.

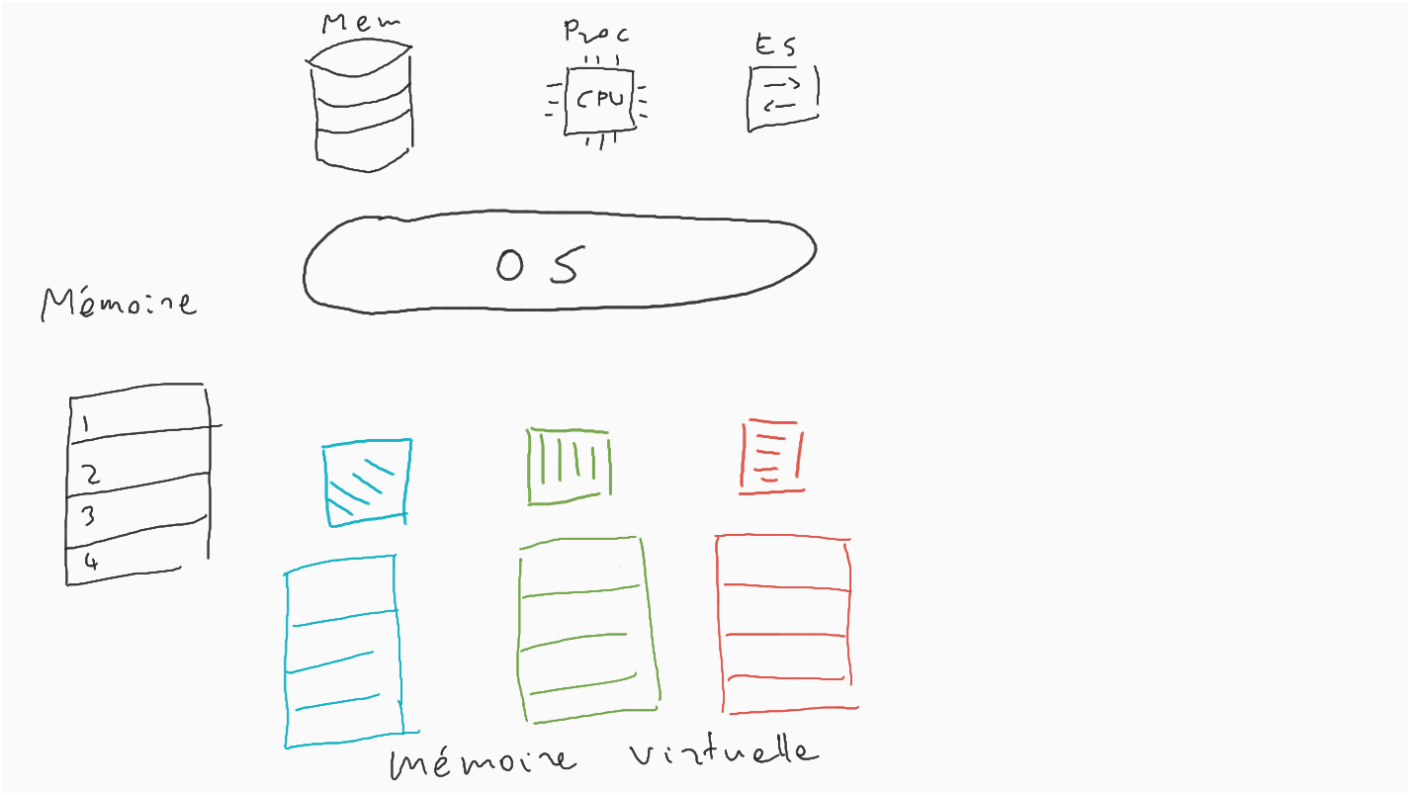


Si l'application verte veut lire ce qu'elle a écrit plus tôt, elle ne va pas récupérer le mot qu'elle a écrit, il est maintenant écrasé ! Ceci occasionne un bug.

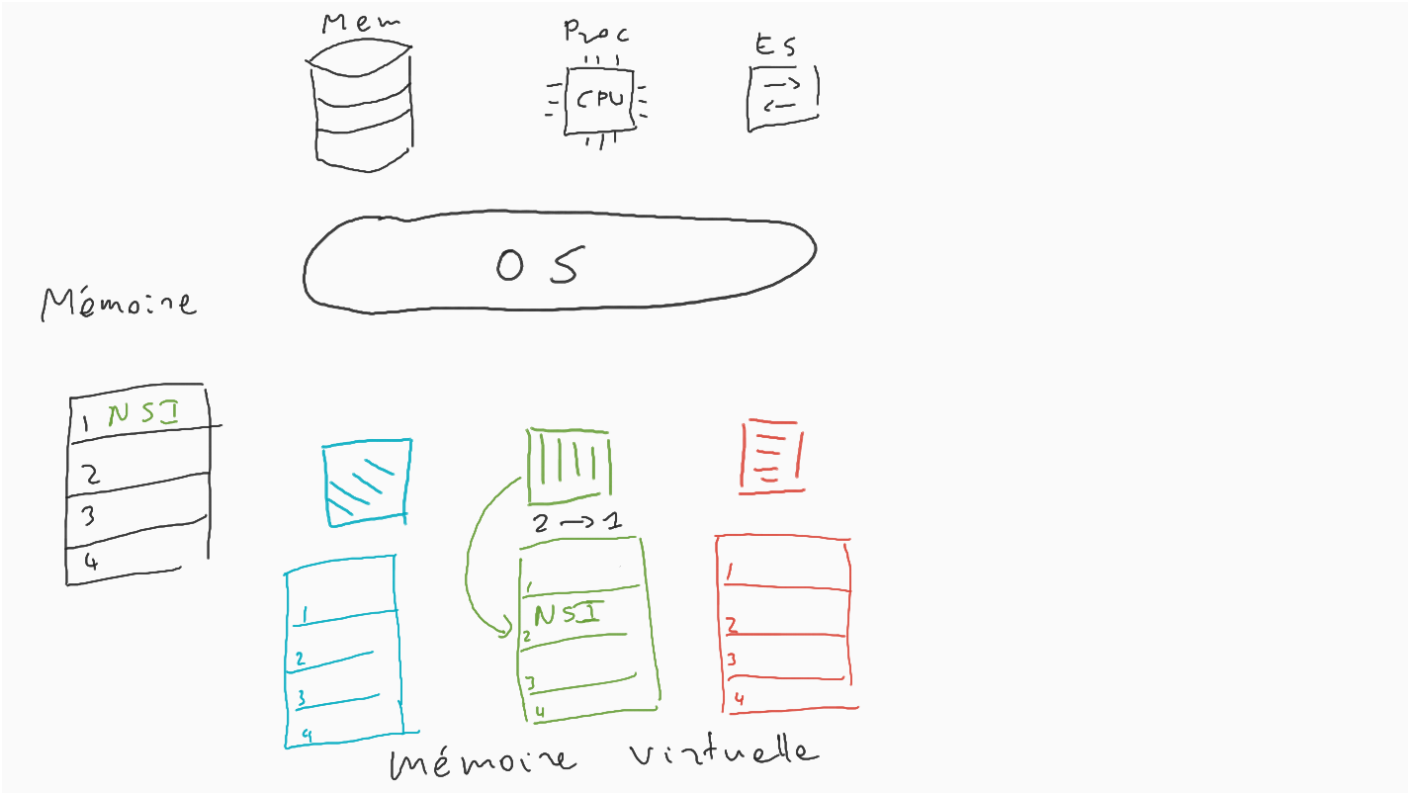


La mémoire virtuelle

Le système d'exploitation se place entre la mémoire physique (qui existe) et les applications. Il donne à chaque application une mémoire virtuelle qui lui laisse croire qu'elle a accès à toute la mémoire de l'ordinateur.



Lorsque l'application verte écrit à l'adresse virtuelle 2 son mot NSI, l'OS copie ce mot dans la mémoire physique à un emplacement disponible, par exemple l'emplacement réel 1.

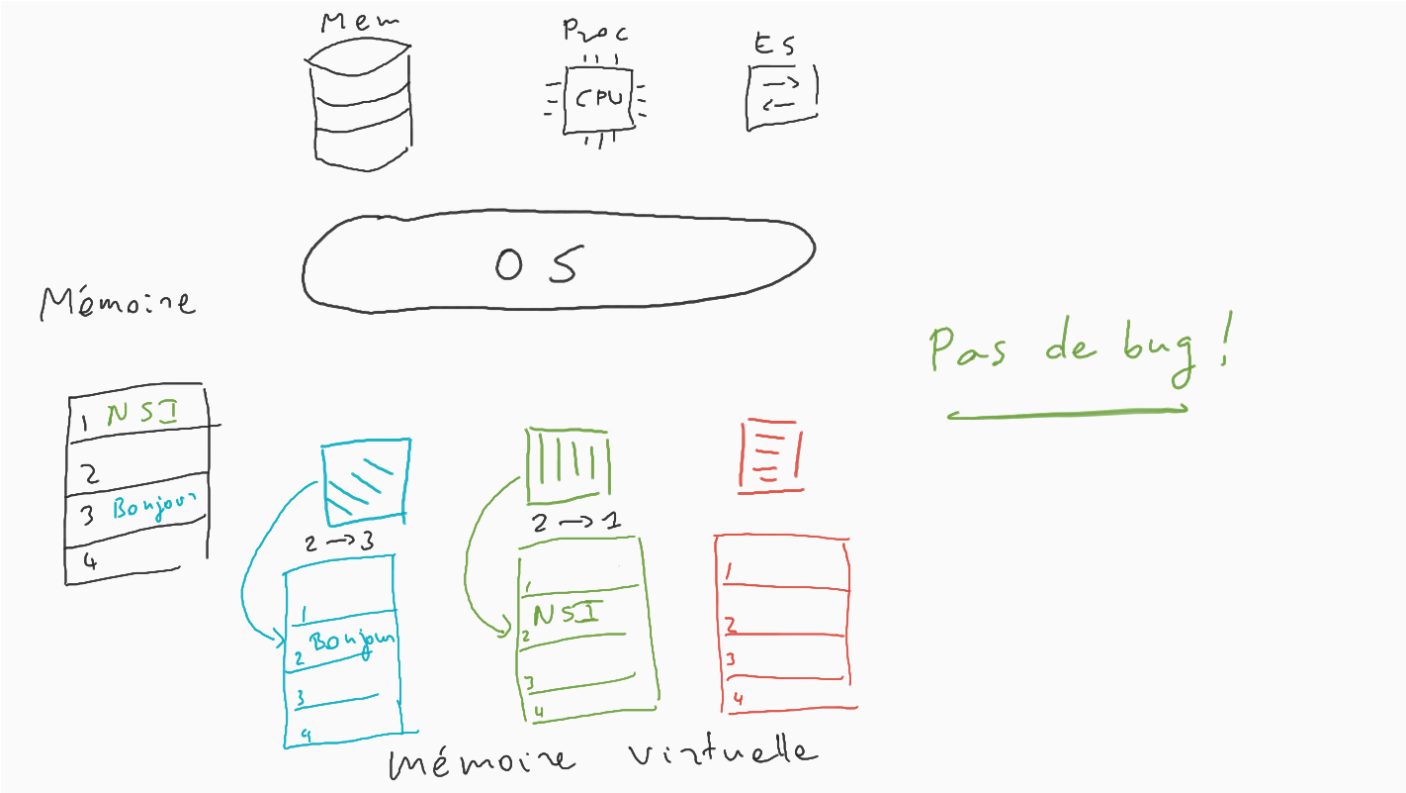


Ainsi, lorsque l'application bleue écrit à l'adresse virtuelle 2 le mot "Bonjour", celui-ci est copié à un emplacement réel disponible, par exemple le 3.

L'OS retient alors que :

- pour l'appli. verte : 2 -> 1,

- pour l'appli. bleue : 2 -> 3.



Lorsque l'appli verte veut consulter le mot 2, l'OS lui renvoie alors ce qu'il a copié à l'emplacement réel 1. Ainsi l'application verte récupère bien le mot "NSI" et le bug est évité.

La mémoire virtuelle est donc un moyen (parmi d'autres) de mettre en commun une ressource (la mémoire) entre différentes applications.

Elle rend l'utilisation multi-tâches possibles : plusieurs programmes peuvent s'exécuter en même temps.

L'ordonnancement des inscructions processeur

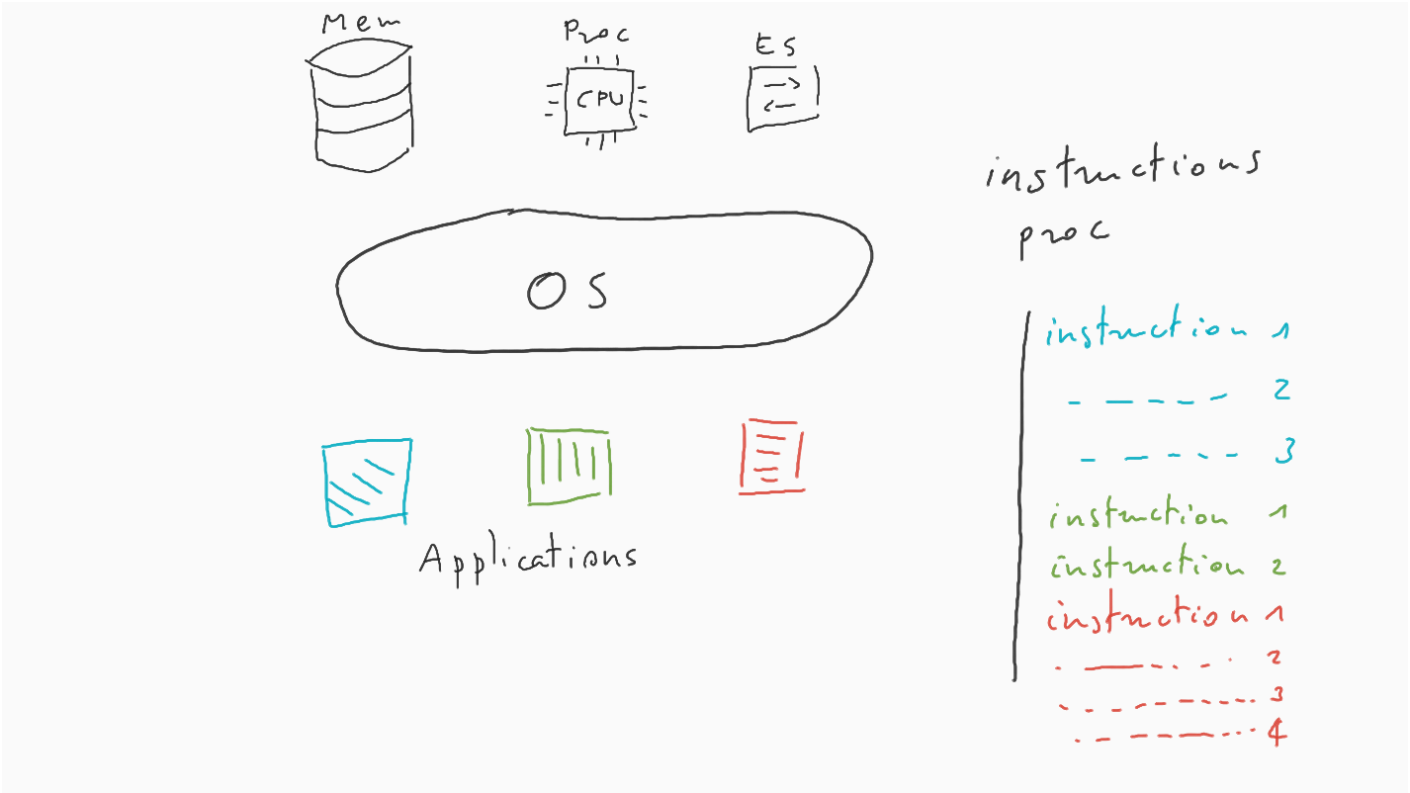
Le processeur, même s'il dispose parfois d'un grand nombre d'instructions possibles ne peut en traiter qu'une à la fois.

Chaque application qui doit réaliser un calcul envoie ceux-ci dans une file d'attente et... pendant ce temps, les applications attendent leur tour.

De la même manière que l'instituteur répond à une question à la fois...

Comment empêcher qu'une application ayant énormément de calculs à faire ne paralyse le système ?

Imaginons demander des milliards d'instructions, si elles sont toutes traitées en une fois, il ne se passera plus rien et la machine sera totalement figée.



Vous ne pourrez même plus bouger la souris à l'écran, les vidéos s'arrêtent… Plus rien tant que les instructions ne sont pas traitées.

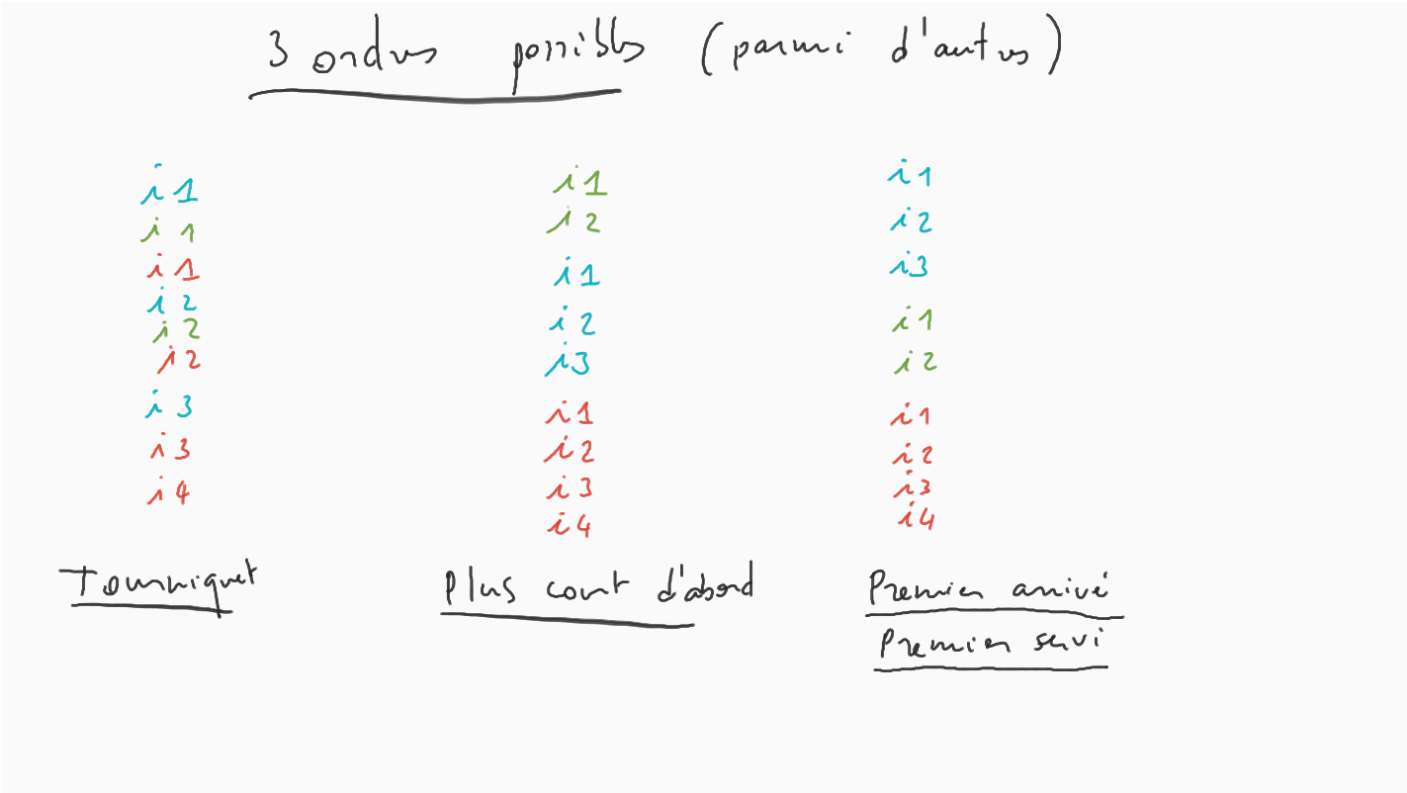
L'ordonnanceur

Le rôle de l'ordonnanceur est de donner la parole à chaque application selon des règles pré-établies et afin de donner l'illusion de la simultanéité.

Ainsi les applications voient leurs instructions se réaliser régulièrement et ne sont pas figées.

Plusieurs solutions sont possibles, par exemple :

- Tourniquet : une instructions de chaque application jusqu'à épuisement,
- Plus court d'abord : traite en premier les instructions de l'application qui en a demandé le moins,
- D'un seul bloc (ce qu'on souhaitait éviter !)



À nouveau, l'application permet d'éviter de paralyser le système et l'ordonnanceur est un autre procédé employé pour rendre le système multi-tâches.

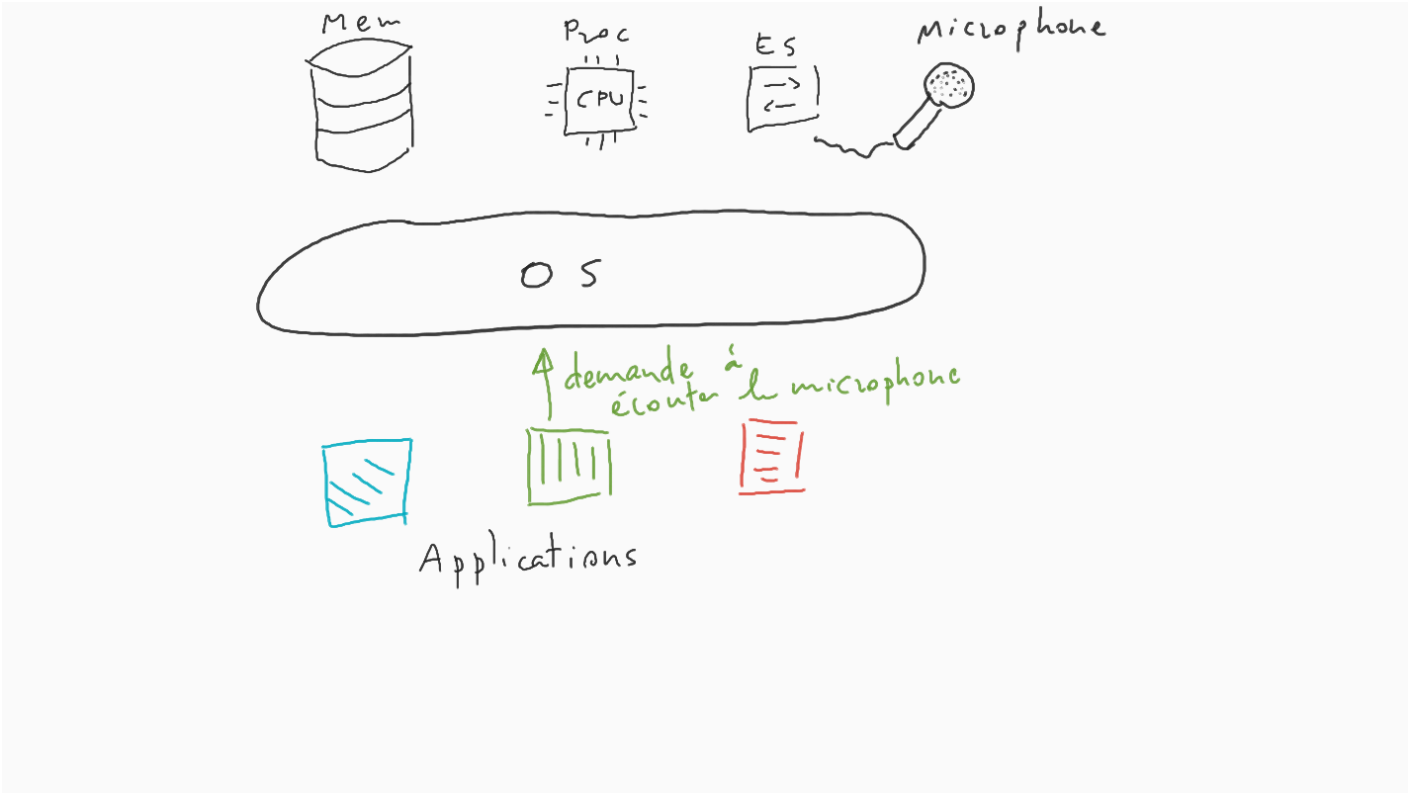
Les interruptions

Comment rendre les périphériques accessibles aux applications… alors qu'on a des centaines de variantes de ces périphériques ?

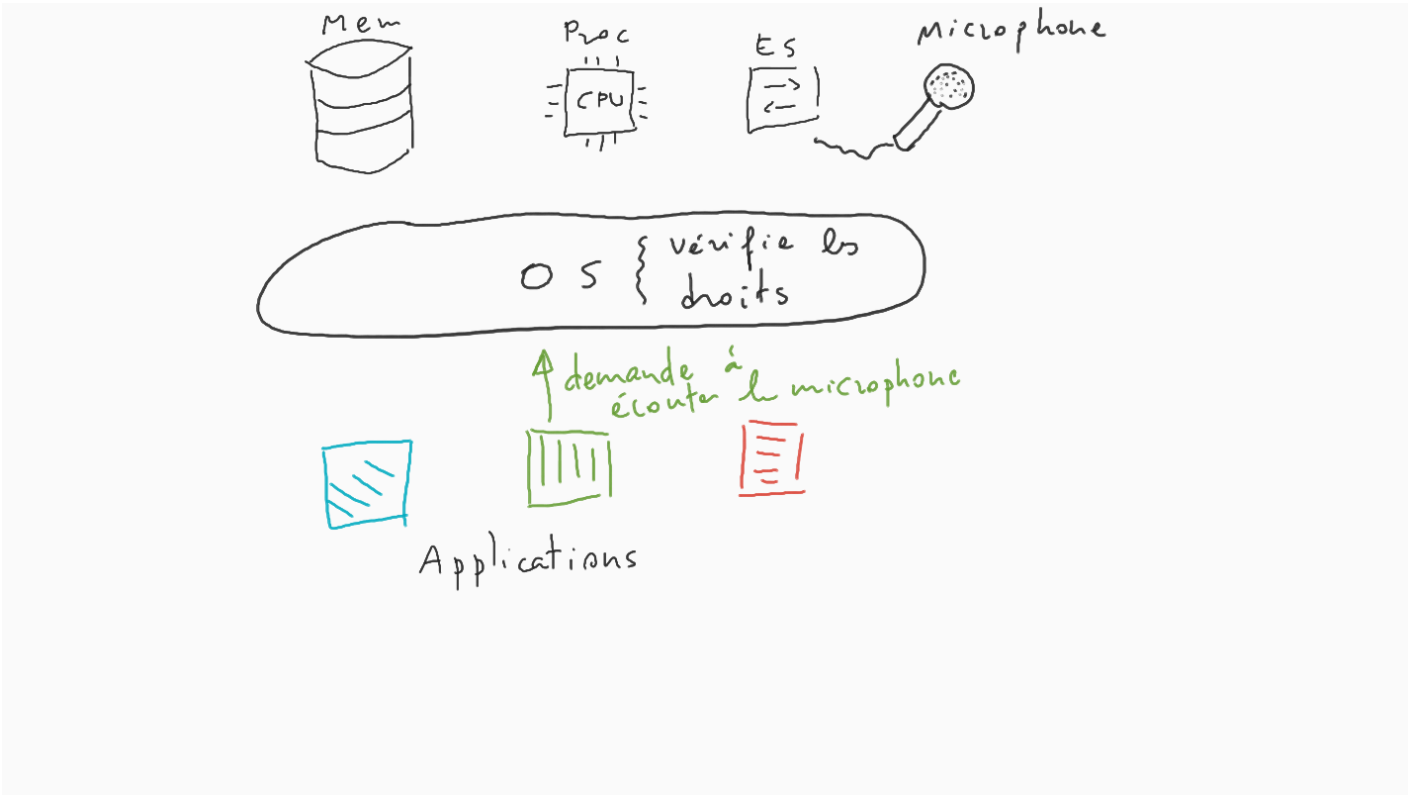
Chaque génération de smartphones dispose de son propre matériel qui fonctionne différemment de la version précédente… Et pourtant, je peux utiliser un logiciel antérieur sur un nouveau téléphone…

Les appels systèmes

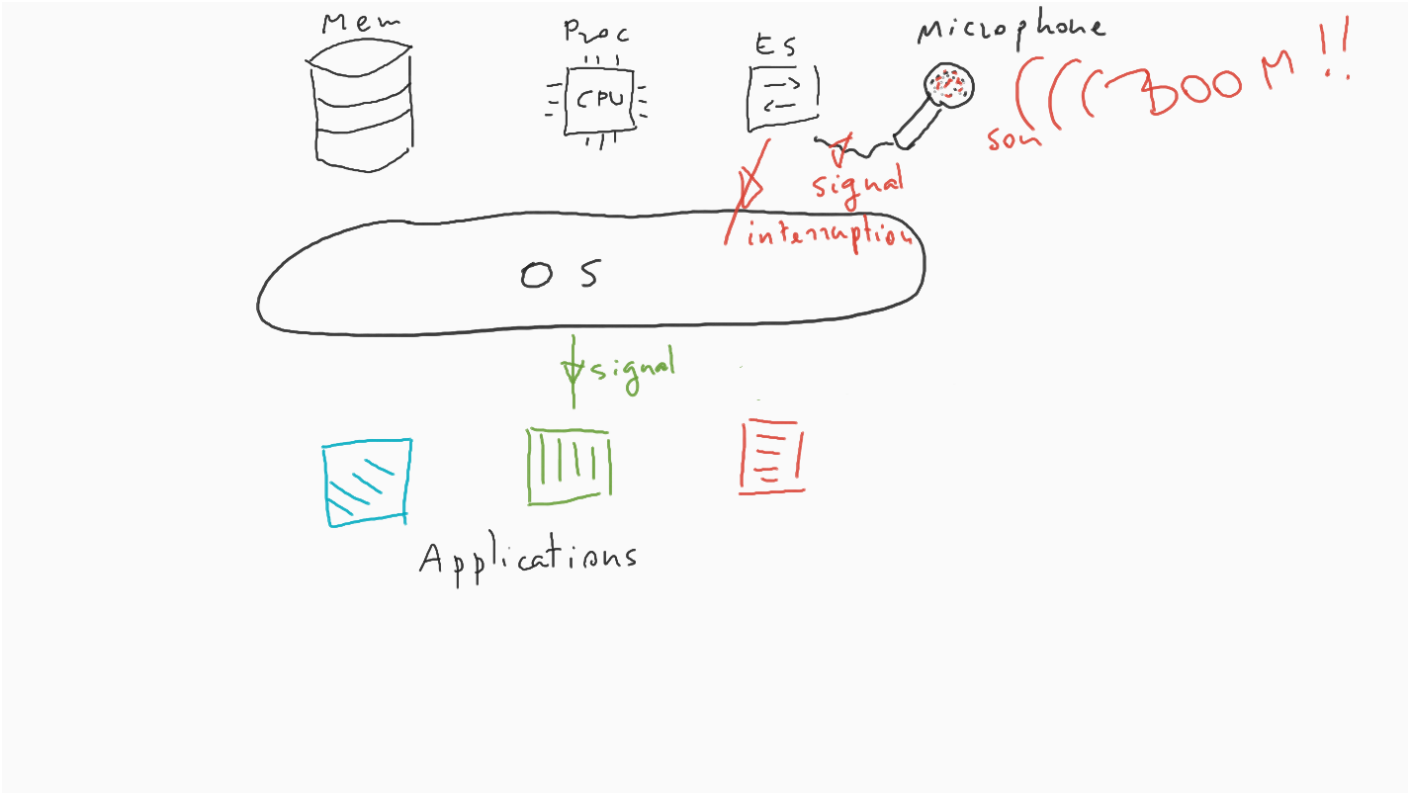
Lorsqu'une application souhaite utiliser le matériel, l'OS commence par vérifier qu'il en a le droit.



Par exemple si l'application souhaite écouter le micro-phone, il faut qu'elle puisse le faire.



Si c'est le cas, l'OS attend qu'un événement venant du micro-phone se produise, ce qu'on appelle une *interruption* et lorsque c'est le cas, elle transmet ce signal à l'application.



Les appels systèmes sont formalisés et sont donc indépendant du matériel.

Un mot sur le matériel : les pilotes

Lorsqu'un fabricant crée un nouveau matériel, compatible avec une machine, on crée alors un *pilote*. C'est d'ailleurs généralement le fabricant lui-même qui le fait.

Celui-ci permet justement de répondre aux appels systèmes standardisés définis par l'API (*Application Programming Interface*) de l'OS.

Ainsi, les matériels très variés communiquent tous de la même manière avec les applications, à travers les pilotes et l'API.

Multitude d'OS

Si tous les OS fonctionnent sur le même principe, pourquoi en existe-t-il autant ?

Parce que certains sont spécialisés :

On n'utilise pas Android sur un super calculateur conçu pour réaliser très vite des milliards de calculs mais une version de Linux spécialement conçue. Et pourtant, ces deux reposent tous les deux sur le même noyau, celui de Linux.

Ensuite, certains OS sont propriétaires, c'est le cas de Windows. Bien qu'il existe des versions gratuites ou d'usage gratuit de Windows, le seul moyen de consulter le code source de celui-ci est de travailler chez Microsoft.

Ce n'est pas le cas des systèmes dits libres dont le code est consultable par tout le monde. Voici par exemple une copie du code source de Linux...

Chose amusante : ce dépôt appartient à Linus Torvald, créateur à la fois de Linux et du logiciel qui fait tourner Github : Git... mais Github (le site lui même) appartient à Microsoft, le grand rival de Linux... Le monde est petit.