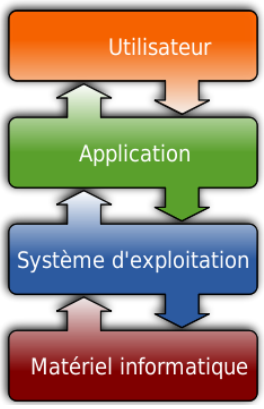
A- Classification des principaux systèmes d'exploitation (Operating System).

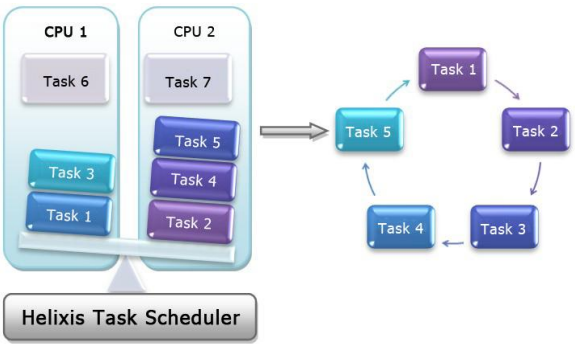
Familles	Sous familles	Développement	Systèmes numériques
Windows	-		Ordinateur personnel (PC) 90% de part de marché
UNIX	Berkeley Software Distribution BSD	MAC OS X	MAC
		iOS	iPhone – iPad – iPod touch
		FreeBSD – Net BSD OpenBSD	Serveur de façon général
	Linux	Androïd	Smartphone -Tablette
		GNU/Linux	Ordinateur personnel (PC)
		Linux	Linux équipe 97% du top 500 des **supercalculateurs** les plus puissants au monde
		Linux	Systèmes embarqués : Box, lecteur de salon, console de jeu, TomTom, baladeur, RaspberryPi.

B- Les différents rôles du système d'exploitation



Le système d'exploitation est la couche entre les applications (programmes utilisateurs) et la partie matérielle (écran, clavier, disque dur, port de communication, ...).
Le système d'exploitation est le premier programme exécuté lors de la mise en marche de l'ordinateur[2], après l'amorçage.
Un système d'exploitation a pour but :

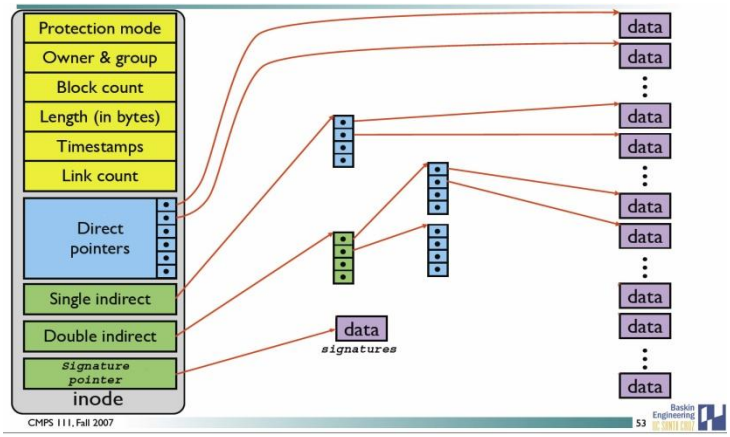
- de **décharger le programmeur** d'une tâche de programmation énorme et fastidieuse, et de lui permettre de se concentrer sur l'écriture de son application,
- de **gérer la bonne exécution des processus**. Il permet à ce titre de « tuer » un processus ne répondant plus correctement,
- de **contrôler l'accès des processus** aux ressources matérielles par l'intermédiaire des pilotes (appelés également gestionnaires de périphériques ou gestionnaires d'entrée/sortie),
- de **gérer la sécurité** liée à l'exécution des programmes en garantissant que les ressources ne sont utilisées que par les programmes et utilisateurs possédant les droits adéquats,
- de **gérer la lecture et l'écriture des fichiers** en fonction des droits d'accès à ces fichiers,
- de **gérer l'allocation du processeur** entre les différents processus grâce à un algorithme d'ordonnancement. Le type d'ordonnanceur est totalement dépendant du système d'exploitation, en fonction de l'objectif visé,
- de **gérer l'espace mémoire** alloué à chaque application et, le cas échéant, à chaque usager. En cas d'insuffisance de mémoire physique, le système d'exploitation peut créer une zone mémoire sur le disque dur, appelée « mémoire virtuelle ». En contrepartie cette mémoire est beaucoup plus lente.



C- Gestion des fichiers (pour les systèmes de type Unix)

Tous les fichiers sont gérés à partir d'un i-nœud qui contient toutes les informations concernant le fichier, sauf son nom. Il existe en fait deux types d'i-nœuds :

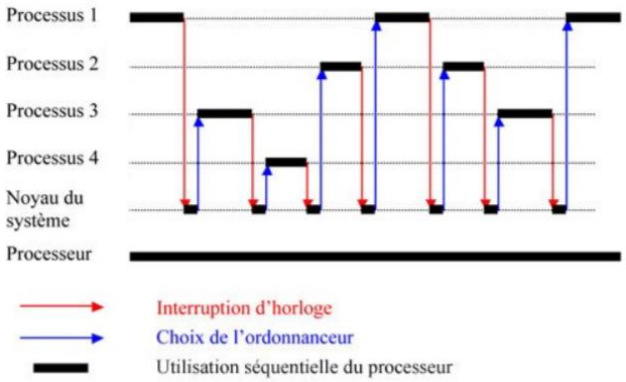
- l'i-nœud sur disque (dans la partie non volatile de la mémoire) : ces i-nœuds sont rangés dans un tableau sur le disque, le numéro d'i-nœud est en fait le numéro de la case dans ce tableau ; ces i-nœuds sont utiles pour conserver des données ;
- l'i-nœud en mémoire (dans la partie volatile de la mémoire) : il contient en gros les mêmes types de renseignements, plus le numéro d'i-nœud associé ; ces i-nœuds sont beaucoup plus rapides d'accès, mais il y en a moins.



Tout le travail se fait en mémoire. Quand l'utilisateur fait une sauvegarde, les modifications sont alors recopiées sur le disque. Supposons qu'un utilisateur veuille ouvrir un fichier avec un éditeur de texte : il le désigne par son nom. Le système trouve alors le numéro d'i-nœud associé en parcourant le répertoire qui contient le fichier. Si l'i-nœud ne se trouve pas en mémoire, le système le charge en mémoire, puis vérifie les droits d'accès. Le système transmet alors un descripteur correspondant à une ouverture du fichier.

D- Les systèmes d'exploitation multitâches

L'utilisateur a l'impression que plusieurs tâches peuvent s'exécuter en même temps. Un processeur ne peut pourtant exécuter qu'un seul processus à la fois, qu'on appelle processus actif. L'ensemble de la mémoire associé à un processus est appelé son contexte (code qu'il exécute, pile d'appel, données, le compteur qui lui permet de savoir où il en est de son exécution, etc.). L'ordonnanceur change régulièrement de processus actif en procédant à un changement de contexte. Ainsi le nouveau processus actif n'a pas conscience qu'il a été interrompu et continue là où il en était. Un des algorithmes classiques d'ordonnancement de processus est appelé round robin. On peut imaginer que les processus sont rangés dans une liste chaînée circulaire et à chaque tic d'horloge l'ordonnanceur donne la main au processus suivant dans cette liste. Si le processus se termine avant le tic suivant, il est supprimé de la liste, sinon il y reste.



E- Systèmes propriétaires vs. systèmes libres

Il existe divers systèmes d'exploitation et tous n'apportent pas les mêmes solutions. Indépendamment des solutions apportées, il existe deux familles de systèmes : les systèmes propriétaires et les systèmes libres.
Pour les systèmes (ou logiciels) libres, le code source est public. On peut en général le modifier ou s'en servir pour fabriquer de nouveaux produits (il est quand même prudent de connaître la licence sous laquelle le logiciel a été publié qui peut préciser ces droits et des obligations du type citer les auteurs originaux ou non, pouvoir fabriquer des logiciels propriétaires ou non, etc.).
Les logiciels propriétaires sont en général non ouverts, il est donc plus difficile (voire illégal) de les modifier.
Les logiciels libres sont souvent maintenus par la communauté, mais peuvent aussi l'être par des entreprises qui les utilisent et qui ont intérêt à ce qu'ils restent efficaces et utilisés par d'autres, ce qui assure l'existence de développeurs susceptibles de participer à leur maintien. Les logiciels propriétaires quant à eux sont essentiellement développés et mis à jour par l'entreprise qui les possède et qui peut décider d'arrêter de les maintenir. Ce sont deux modèles économiques très différents.