

même architecture matérielle. Cette vidéo en donne une introduction claire.

2 Système d'exploitation

En usage courant d'un ordinateur — quel qu'il soit —, lors du lancement d'une application de lecture de vidéo, les images apparaissent à l'écran et du son est émis par les haut-parleurs. Dans le même temps, le gestionnaire de courrier électronique peut se manifester en prévenant de l'arrivée d'un nouveau message, auquel il est possible de répondre en saisissant du texte au clavier sans avoir à quitter le lecteur vidéo — et, en toute convenance pratique, sans obliger à reprendre la lecture depuis le début alors que le dénouement est proche!

Comment une application peut-elle ainsi interagir avec les périphériques matériels? Comment deux applications peuvent-elles tourner en même temps sur un seul processeur? Tout ceci est possible grâce au *système d'exploitation*, une couche logicielle intermédiaire entre la couche applicative et celle du matériel, dont les principes de fonctionnement sont à découvrir en trois concepts clefs.

À PROPOS DE L'INTERVENANT

Damien SAUCEZ est chercheur au sein de l'équipe INRIA-DIANA. Il consacre ses travaux aux réseaux information-centrés (*Information-centric networking*) — par exemple, les problèmes de routage et de contrôle de la congestion —, aux réseaux définis par logiciel (*Software Defined Networking*) — en particulier, les questions de résilience et de robustesse — et aux expérimentations à large échelle. Il est un contributeur actif de l'ietf et de l'irtf.

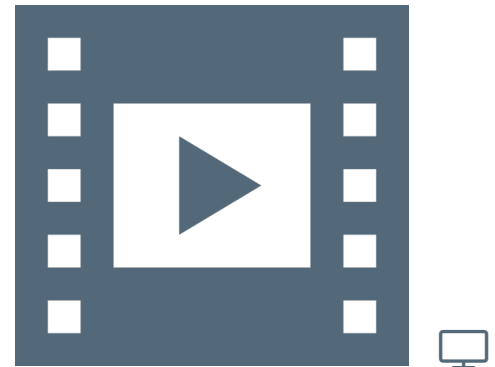
2.1 Système d'exploitation en trois idées-forces

WINDOWS, LINUX, MACOS, ANDROID, IOS ou autre, sont des noms régulièrement entendus correspondant à une utilisation quotidienne mais, au final, à quoi cela fait-il vraiment référence?

Ce sont en fait des logiciels spécialisés qui sont appelés *systèmes d'exploitation*. Un OS/*système d'exploitation* sert dans sa fonction première à gérer le lien entre tout ce qui appartient aux mondes applicatifs — autrement dit les logiciels utilisés — et le matériel à disposition tels que les processeurs, les disques durs, les écrans et ainsi de suite...

Il faut se souvenir qu'un ordinateur est essentiellement composé de trois éléments principaux : la *mémoire*, — qui sert à stocker de l'information et à y accéder quand on en a besoin —, le ou les *processeur(s)* — cerveau(x) de l'ordinateur, il(s) exécute(nt) les calculs et les tâches sur la base des informations qu'il(s) récupère(nt) en mémoire — et enfin, pour communiquer avec le monde extérieur, les *périphériques d'entrée/sortie* qui prennent en compte une commande ou restituent un résultat, qu'il s'agisse d'une imprimante, d'un écran, d'un clavier ou d'une souris, etc. Dans cette perspective, Internet peut également être considéré comme un périphérique d'entrée et de sortie.

Bien entendu, un ordinateur fait tourner plusieurs *applications* en même temps; chacune d'entre-elle effectuant une tâche bien précise. Afin d'utiliser le matériel, on ne va pas les autoriser à accéder directement aux différents composants de l'ordinateur. C'est le rôle du système d'exploitation — *Operating System* en anglais — que d'être l'intermédiaire entre *application* et matériel et d'établir le lien entre les différents éléments de l'ordinateur. Ces derniers se déclinent en trois catégories principales : la mémoire, les processeurs et les entrées-sorties.



VIDÉO 1.3 — Système d'exploitation.

2.1.1 Gestion de la mémoire

Pour se faire une idée, la mémoire peut s'imaginer comme un grand tableau disposant de plusieurs entrées que l'on appelle des mots. Chaque mot correspond à une *adresse* de la mémoire physique.

Par exemple, en supposant que quatre mots sont à écrire en mémoire, ils vont se référer aux adresses numérotées de 0³³ à 3. En pre-

33. Dans la réalité c'est un peu plus complexe, mais le principe est identique.



FIGURE 1.7 — Principe de fonctionnement de la mémoire virtuelle.

mière approche, on pourrait alors considérer qu’une application qui souhaite enregistrer le mot « toto » en mémoire physique, lui attribue par exemple l’adresse n°1. Une fois écrit en mémoire, l’application peut y accéder. En imaginant qu’une deuxième application veuille écrire le mot « titi » elle aussi à l’adresse n°1, cela va écraser le mot « toto » pour le remplacer par « titi ». Il y a clairement un problème, car la première application a perdu le mot dont elle a besoin et récupère à la place un mot inconnu. Cela ne fonctionne pas.

Le système d’exploitation permet d’éviter ce genre de déconvenues grâce au concept de *mémoire virtuelle*. L’idée sous-jacente est que le système d’exploitation va donner à chaque application l’illusion qu’elle dispose de toute la mémoire. La mémoire virtuelle s’apparente donc toujours à un tableau de mots avec leurs adresses respectives. En suivant l’exemple précédent, l’application va donc, de son point de vue, toujours écrire le mot « toto » à l’adresse n°1, mais en mémoire virtuelle. Quant à lui, le système d’exploitation est en charge de trouver un espace où écrire le mot dans la réelle mémoire physique et, bien entendu, de transcrire le lien entre mémoire virtuelle et mémoire physique. Dès lors, il n’y a plus de conflit de mémoire et de problème d’inconsistance entre les différentes applications lancées : ce qui va être l’adresse n°1 pour chaque application considérée, correspond en fait à *plusieurs adresses physiques*.

2.1.2 Gestion du processeur

Un ordinateur contient au moins un processeur et, s’il en possède plusieurs, leur nombre est toujours largement inférieur à celui des applications qui sont potentiellement à faire tourner conjointement.

Le système d’exploitation sert d’arbitre entre les diverses applications, de manière à partager « équitablement³⁴ » l’accès au(x) processeur(s). En effet, un processeur est « stupide » ; il se contente uniquement d’effectuer des calculs³⁵ et de ne réaliser qu’une tâche à la fois. C’est un peu comme dans un débat, la parole est concédée chacun son tour par le modérateur. Le système d’exploitation agit de manière similaire par l’intermédiaire de l’*ordonnanceur*. L’ordonnanceur établit ainsi qu’une application a assez effectué de calculs pour la mettre en pause et offrir l’accès au processeur aux autres applications.

Il existe néanmoins des règles d’optimisation. Par exemple, étant donné que les temps d’accès aux disques durs sont relativement plus importants que ceux des mémoires caches ou de la mémoire vive (RAM), au lieu d’attendre l’aller-retour avec le disque dur pour récupérer une information nécessaire à l’algorithme de calcul — et perdre ainsi un temps précieux —, le système d’exploitation place en réserve l’application en cours pour libérer le processeur et exécuter une autre application avant de reprendre le traitement une fois l’information obtenue.

2.1.3 Entrée-sortie : appels système

La typologie des ordinateurs est très disparate : un téléphone mobile, une carte à puces, un distributeur de billets, une station personnelle de bureautique, un *cluster* de calcul — simulation météo, automobile ou aéronautique —, un serveur de *Data center*, etc.

L’enjeu principal est de pouvoir écrire ou plus exactement programmer des applications qui, dans la mesure du possible, possèdent le même code de base pour chaque configuration matérielle sous-jacente, sans avoir besoin de la connaître. On parle de développement multiplateforme. Bien que tout ne soit pas toujours possible aussi directement, les choses ont considérablement évoluées ces dernières décennies. Cependant, dans nombre de cas d’utilisations courantes, cela

34. Il existe une hiérarchie dans les applications, que ce soit en termes de sécurité, de priorité ou autre critère prépondérant.

35. Certes, les processeurs sont suffisamment denses en circuits imprimés, donc véloces pour donner l’impression qu’ils réagissent simultanément ou « en temps réel » aux instructions. Que *nenni* !

s'avère envisageable *via* le système d'exploitation, lequel fournit une couche d'abstraction entre matériel et logiciel : les *appels systèmes*.

Quid des appels systèmes ? À chaque fois qu'une application a besoin d'exécuter une tâche, comme déjà vu, elle ne va pas directement accéder au matériel, mais formuler et déléguer au système d'exploitation sa demande d'intervention. Par exemple, dans le cas d'un jeu, l'application veut afficher un rectangle³⁶ à l'écran. Le système d'exploitation va d'abord vérifier que l'application est autorisée à le faire puis, dans l'affirmative, examiner comment opérer et enfin, réaliser la tâche en tant que telle en dessinant le rectangle voulu à l'écran.

Autre exemple, un microphone. Une application souhaite enregistrer le son fourni par un microphone connecté à l'ordinateur. Une fois la requête émise, le système d'exploitation s'attache à « ouvrir le canal microphone » et se met à l'écoute. Lors de l'apparition d'un événement sonore, le système d'exploitation génère une interruption, enregistre le signal d'entrée et le redirige sur l'application demandeuse.

36. Dans la pratique et au delà des jeux, les systèmes d'exploitation ne sont pas forcément associés à une interface graphique. C'est par exemple le cas des serveurs : une interface graphique apportant potentiellement des failles de sécurité. Néanmoins, les autres configurations à vocation d'interaction avec un utilisateur incluent une interface graphique proche du système d'exploitation : on parle alors de bibliothèque graphique associée. À titre d'illustration, pour les systèmes GNU/LINUX, il existe deux interfaces majeures : GTK+/GNOME et QT/KDE.

GLOSSAIRE CONTEXTUEL

OS/SYSTÈME D'EXPLOITATION — OS, acronyme de *Operating system*, est l'expression anglophone signifiant « système d'exploitation ». Il s'agit d'un ensemble de programmes qui gèrent l'utilisation des ressources matérielles d'un ordinateur : stockage des mémoires à accès rapide et de masse des disques durs, ordonnancement des calculs du processeur, communication avec les périphériques ou administration du réseau. Le système d'exploitation accepte ou refuse ces demandes, puis réserve les ressources en question pour éviter que leur utilisation n'interfère avec d'autres requêtes provenant d'autres logiciels.

APPLICATION — Une *application* ou un *applicatif* est, dans le domaine informatique, un programme — ou un ensemble logiciel — directement manipulé par l'utilisateur pour réaliser une tâche ou un ensemble de tâches élémentaires d'un même domaine formant un tout. Typiquement, un éditeur de texte, un navigateur Web, un client de messagerie, une visionneuse d'images, un lecteur multimédia, un jeu vidéo, sont des applications.

MÉMOIRE — Dispositif électronique qui sert à stocker des données (valeurs, instructions de programmes). Dans un ordinateur, il existe trois grands types de mémoire externe au processeur : la mémoire vive (RAM) (carte branchée sur la carte mère), la mémoire morte (ROM) (circuit intégré à la carte mère) et la mémoire de masse (disques durs, et autres supports de stockage périphérique). Le processeur intègre aussi plusieurs niveaux de mémoire, notamment de la mémoire cache (placée dans la puce qui contient le processeur) et les registres (internes au processeur).

MÉMOIRE VIRTUELLE — Mécanisme qui permet de simuler la présence d'un type de mémoire en utilisant un autre type (par exemple un disque dur). Il est utilisé par exemple pour simuler la présence de mémoire vive en utilisant de la mémoire de masse. Sous LINUX, ce type de mécanisme est dénommé « *swap* » où à l'installation du système d'exploitation, une partition du disque dur est réservée à cet effet (on conseille souvent une taille double de celle de la mémoire vive de l'ordinateur).

PROCESSUS — Forme que prend un programme quand il est exécuté au sein d'un système d'exploitation. Une instance de proces-

sus comprend en général : un ensemble d'instructions (souvent copiées dans la RAM depuis le disque dur), une place, appelée *espace d'adressage*, réservée en mémoire vive pour stocker les données qu'il manipule et les ressources matérielles et logicielles que le programme utilise.

ORDONNANCEUR — Composant logiciel du système d'exploitation qui est en charge d'allouer du temps processeur aux processus. Le choix du processus élu pour s'exécuter sur le processeur à un temps donné est fait par une *procédure d'ordonnement* et implique de gérer les *changements de contexte*.

EN BREF...

Ainsi, le système d'exploitation gère la mémoire au moyen de la mémoire virtuelle, priorise l'accès au processeur entre les différentes applications concurrentes et abstrait les détails techniques de la machine pour avoir des codes d'application les plus transparents possibles vis-à-vis du matériel constituant l'ordinateur; d'où, notamment, la notion et la présence de pilotes de périphérique — ou *drivers* en anglais —, proches du noyau du système d'exploitation et à considérer comme extension, qui s'avèrent nécessaires au fonctionnement des matériels non supportés nativement par le système d'exploitation.

NOTE DE LA RÉDACTION

Texte rédigé par Claude KAISER et publié sur *Interstices* — revue en ligne de culture scientifique du numérique — le 23 juin 2015.

2.2 À quoi sert un système d'exploitation ?

On se moque aisément de Monsieur JOURDAIN qui faisait de la prose sans le savoir. Mais en utilisant un ordinateur ou une tablette, se déclenche sans le savoir tout un flot de programmes souterrains, ceux du système d'exploitation, dont les rôles importants demeurent cachés.

L'informatique est une science jeune, comportant beaucoup d'aspects techniques et une forte compétition industrielle. Aujourd'hui, il existe un système d'exploitation dans chaque ordinateur : tablette, téléphone portable et plus généralement tout objet numérique, avec sur le marché des dizaines de systèmes d'exploitation différents. Dans leur nom, on trouve souvent le sigle OS pour *Operating System* en anglais. Même si les constructeurs utilisent des noms différents pour se démarquer de leurs concurrents, on retrouve dans tous les systèmes d'exploitation des aspects communs et des invariants.

D'un ordinateur ou d'une tablette, on distingue d'abord l'écran et le boîtier qui enferme le matériel — processeurs, mémoires et canaux d'entrées-sorties —, mais c'est le logiciel qu'il contient que l'on utilise.

La programmation directe de l'ordinateur est laborieuse. La machine, construite sur le modèle de VON NEUMANN, est universelle (voir sur *Interstices* « *Sous le signe du calcul* », par Jean-Louis GIAVITTO et François RECHENMANN et « *Alan TURING : du concept à la machine* », par Sacha KRAKOWIAK). Toutefois, le langage qu'elle reconnaît, composé d'instructions et de données, toutes codées en binaire, est très difficile d'emploi. Le système d'exploitation vient alors au secours de l'utilisateur, en lui permettant d'exploiter l'ordinateur mieux qu'en langage machine. Mais il fait plus que cela.

A. RÔLES DU SYSTÈME D'EXPLOITATION

Le but d'un système d'exploitation est de rendre aisée l'utilisation de l'ordinateur par chacun, comme s'il s'agissait d'une machine fictive, sa « machine virtuelle », qui aurait été construite pour lui. Le système fournit un accès commode et ergonomique, par exemple avec un écran comportant des fenêtres multiples et une interface graphique. De nos jours, en 2015, il peut aussi y avoir une interface tactile ou sonore.

Le système assure ainsi le stockage de programmes et de données de toutes sortes — textes, images, vidéos, films — dans des fichiers pré-



Source : Flickr CC BY-ND 2.0

Manchot : la mascotte de GNU/LINUX, système d'exploitation libre populaire.

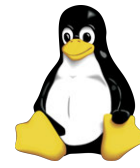
parés par l'utilisateur ou chargés depuis des supports externes ou *via* le réseau. Idéalement, ce premier rôle de gestion de l'information et des fichiers doit permettre à l'utilisateur de n'avoir à préciser que les aspects logiques de son travail.

Le second rôle du système, le contrôle d'exécution, consiste à gérer au mieux les ressources — matérielles et logicielles — qui sont nécessaires pour lancer et suivre l'exécution des applications locales ou distantes. Le système s'occupe ainsi des aspects technologiques et des contraintes d'utilisation des ressources partagées, qu'elles soient attribuées à tour de rôle, comme le sont le processeur et l'imprimante ou divisées, comme le sont la mémoire et l'écran.

Ces deux rôles doivent être pérennes. Pour cela, le système d'exploitation assure — et c'est là son troisième rôle — la sécurité de fonctionnement en cas de panne d'origine interne (matérielle ou logicielle) ou d'agression provenant d'un environnement de plus en plus ouvert (le réseau). Il sauvegarde automatiquement les travaux en cours et veille à permettre le redémarrage après une panne. Il doit aussi rendre possible une évolution matérielle (changement dans la configuration matérielle) ou fonctionnelle (mise à jour et ajout de programmes).

Ces rôles sont remplis par des services décrits dans les programmes du système d'exploitation; l'exécution de ces programmes — et donc des services — est accomplie par ses processus.

Un programme est une entité passive, décrivant une suite d'instructions. Un processus est son pendant dynamique, une entité active qui représente l'exécution de cette suite d'instructions par l'ordinateur.



B. REPRÉSENTATIONS DU SYSTÈME D'EXPLOITATION

Dans sa mouture générique, un système d'exploitation peut s'appréhender selon deux aspects :

- une vision « statique » qui correspond à l'empilement hiérarchique de ses programmes;
- une perspective « dynamique » qui rend compte de l'exécution des programmes par les processus.

REPRÉSENTATION STATIQUE : LES PROGRAMMES — Le système d'exploitation d'un ordinateur est un fantastique regroupement de programmes et de données qui ont été élaborés pour fournir les services requis pour chacun des rôles cités précédemment. Ces informations sont empilées astucieusement en couches qui reflètent la forte structuration de la construction : au niveau le plus bas, on trouve les programmes du noyau qui matérialisent les principes sur lesquels la construction du système est fondée, puis les programmes qui pilotent l'accès au matériel; au-dessus, dans l'ordre ascendant, utilisant parfois les programmes des couches qui leur sont sous-jacentes, viennent les services propres du système d'exploitation, les services communautaires et, enfin, les programmes d'application destinés à l'utilisateur. La figure 1.8 schématise la représentation habituelle de ces services.

Cet empilement structuré d'informations forme une grande base de données : les fichiers du système. Il faut y ajouter les fichiers des utilisateurs entreposés sur des supports gérés par le système d'exploitation ou accessibles à distance sur le réseau.

La rapidité des processeurs et la grande capacité de la mémoire permettent de construire des systèmes très complexes et de taille considérable qui peut atteindre une dizaine de *giga*octets. Par exemple, un système d'exploitation utilisé pour un ordinateur portable, le système d'exploitation MacOSX 10.6, occupe environ 5 Go en mémoire, auxquels s'ajoutent de 5 à 15 Go pour les bibliothèques partagées.



FIGURE 1.8 — Représentation statique du système d'exploitation : descriptif et approche en couches (cf. *Interstices*).

ATTENTION!

Lien mort : https://interstices.info/jcms/c_19937/octet.

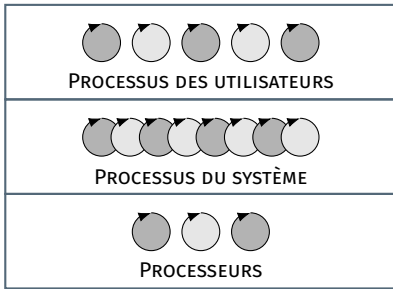


FIGURE 1.9 — Représentation dynamique du système d'exploitation : descriptif et approche en processus (cf. *Interstices*).

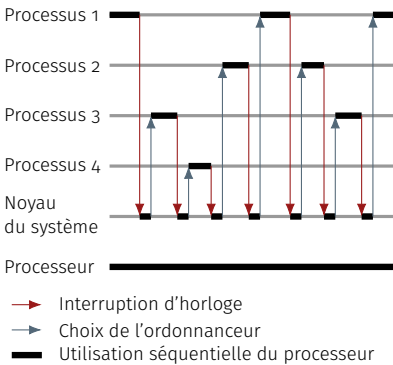


FIGURE 1.10 — Processeur partagé, prêté successivement à quatre processus (cf. *Interstices*).

COMPLÉMENT DE LECTURE
 Pour comprendre dans le détail la dynamique d'un système d'exploitation et découvrir les processus qui se déclenchent lors du lancement d'un ordinateur, le lecteur est invité à consulter l'article « *Le ballet des processus dans un système d'exploitation* », *Interstices*, 2015.

REPRÉSENTATION DYNAMIQUE : LES PROCESSUS — De nombreux programmes du système qui ont leurs sources — codes et données — dans ces empilements de couches sont appelés à s'exécuter de façon concomitante. Ils répondent en temps réel aux besoins explicites ou implicites des utilisateurs et réagissent aux événements aléatoires issus de l'environnement. Quelquefois, c'est le même service qui est demandé simultanément par plusieurs événements ou utilisateurs. Le système d'exploitation peut ainsi être amené à gérer l'exécution concurrente de plusieurs centaines de programmes. Le mot concurrent, polysémique, évoque à la fois la simultanéité — les programmes courent ensemble — et la compétition dans l'attribution des ressources.

Afin de repérer ces différentes exécutions concomitantes, le terme de processus a été introduit. Ce mot a été utilisé dès 1960 dans le système MULTICS. Il sert à identifier le déroulement d'un programme séquentiel — parmi d'autres — et le distingue du texte du programme.

La figure 1.9 schématise cette vision dynamique, qui traduit l'activité des processus. On y distingue les processus du système, à savoir les processus chargés des infrastructures et des services communautaires et les processus créés plus spécialement pour fournir une « machine virtuelle » à chaque utilisateur. Chacun des processus est une abstraction du processeur physique.

C. PROCESSUS ET PROCESSEURS

La représentation dynamique fait apparaître une nuée de processus, alors qu'il n'y a pas assez de processeurs physiques pour en attribuer un à chacun. Or les processeurs physiques ont une vitesse environ un million de fois supérieure à la capacité de réaction de l'utilisateur. Cette énorme différence est mise à profit pour simuler le parallélisme de déroulement des programmes en partageant les processeurs entre les processus — appelé le pseudo-parallélisme. Le partage est rythmé par un top d'horloge qui déclenche l'interruption du processus en cours et l'attribution du processeur à un autre processus. Cette attribution est régie par un programme du noyau du système, l'ordonnanceur.

La figure 1.10 montre un exemple de partage d'un processeur (on suppose ici qu'il n'y en a qu'un) entre quatre processus déclenchés simultanément qui l'utilisent l'un après l'autre, par tranches séquentielles successives. L'attribution du processeur par le noyau consomme aussi du temps de processeur.

D. BILAN

La complexité des systèmes d'exploitation est grande, elle résulte tant du nombre et de la taille des programmes impliqués, que de la multiplicité des processus et de leurs interactions. Cette grande complexité rend illusoire le « zéro défaut »!

Aujourd'hui, on continue de développer de nouveaux systèmes, tout en recherchant l'augmentation de leur sûreté de fonctionnement. Pour cela, on commence à utiliser des techniques automatiques de certification de programmes.

POUR ALLER PLUS LOIN...

MÉTAPHORE BADINE

- ▶ De votre boulangerie à un système d'exploitation multiprocesseur, par Brice GOGLIN, *Interstices*, 15 février 2011.

QUELQUES ASPECTS HISTORIQUES

- ▶ Naissance des systèmes d'exploitation, par Sacha KRKOWIAK et Jacques MOSSIÈRE, *Interstices*, 5 avril 2013;