

Architecture et Système

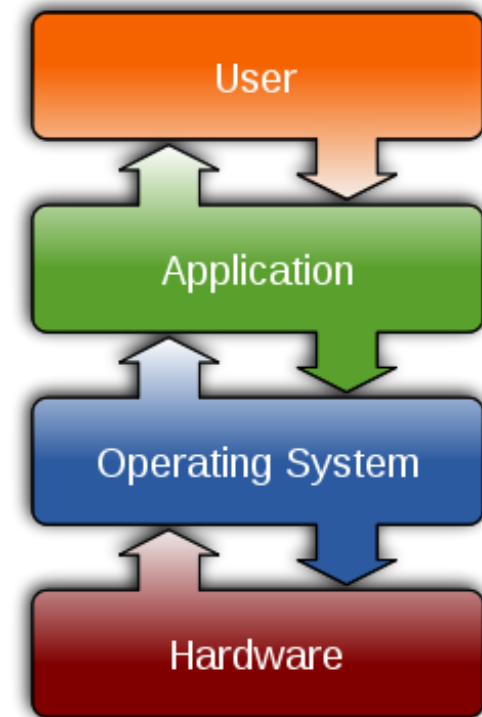
Stefan Schwoon

Cours L3, 2026/2027, ENS Paris-Saclay

Système d'exploitation

Caractérisation selon *Tanenbaum*:

Software consists of two categories: the system programs, which enable the operation of the computer, and the application programs, which resolve the user's problems. The operating system is the most important of the system programs. It controls the resources of a computer and provides a basis for the application programs.



Du coup le système intervient entre le matériel et les logiciels avec lesquels un utilisateur travaille.

Tâches d'un système d'exploitation

Facilitation, par exemple :

Mise à disposition des fonctionnalités communes

Abstraire les détails du matériel, permettre aux mêmes logiciels de fonctionner sur divers ordinateurs (**bibliothèques**)

Permettre la concurrence et communication entre applications (**ordonnancement, gestion de mémoire, entrée/sortie**)

Stockage de données (**système de fichiers**)

Tâches d'un système d'exploitation

Contrôle, par exemple :

Assurer qu'aucun utilisateur ne gêne un autre (par accident ou malveillance)

Protéger les applications d'un utilisateur en cas l'une parmi elles produit une erreur

Assurer une distribution juste des ressources

Protection contre les virus etc.

Modes utilisateur et privilégié

Les processeurs courants typiquement possèdent au moins deux modes différents : **mode utilisateur** et **mode privilégié** (ou *mode noyau*).

Réalisation technique : un bit dans le registre de statut, pris en compte pour certains opérations.

Limitations en mode utilisateur (exemples)

- toute interaction avec les périphériques est interdite

- l'accès mémoire est restreint ; certaines adresses sont 'lecture seulement' ou 'hors limite'

- on ne peut pas interdire les interruptions

- pas de modification des vecteurs d'interruption

Modes utilisateur et privilégié

En mode privilégié :

toute opération est permise

accès mémoire non-restreint

modification de la configuration mémoire (p.ex. droits d'accès)

interdiction des interruptions possible (protection d'opérations critiques)

modification des vecteurs d'interruption

Modes utilisateur et privilégié

L'ordinateur **démarre en mode privilégié** et charge le système d'exploitation.

Le système fait alors le suivant (toujours en mode privilégié) :

- Mise en place du code privilégié (le **noyau**).

- Basculer les vecteurs d'interruption vers les bonnes adresses dans le système (notamment interruption horloge pour l'ordonnancement)

- Configurer les accès mémoire pour les processus utilisateur

- Basculer en mode utilisateur** et lancer un premier processus utilisateur (p.ex. un login)

Modes utilisateur et privilégié

Comment basculer du mode utilisateur vers mode privilégié ?

Réalisation technique selon type de processeur, p.ex.:

- par **interruption** (matériel ou logiciel)

- par une instruction **syscall**

Dans les deux cas, la fonctionnalité du système ...

- est fournie dans une zone mémoire contrôlée et non-modifiable ;

- est le seul code à fonctionner en mode privilégié ;

- fournit un accès indirect aux périphériques etc à travers une interface bien définie.

Les appels système

Interface pour demander un service au noyau :

Mécanisme typique : (**direct.s**)

tout service correspond à un numéro

l'identifiant du service et d'autres paramètres sont mis dans les registres

on exécute un appel système (p.ex. une interruption logiciel)

Ces détails dépendent de la version précise du système – pas standardisé.

Le programmeur n'est pas censé utiliser ce mécanisme directement, mais à travers des bibliothèques.

Organisation d'un système d'exploitation

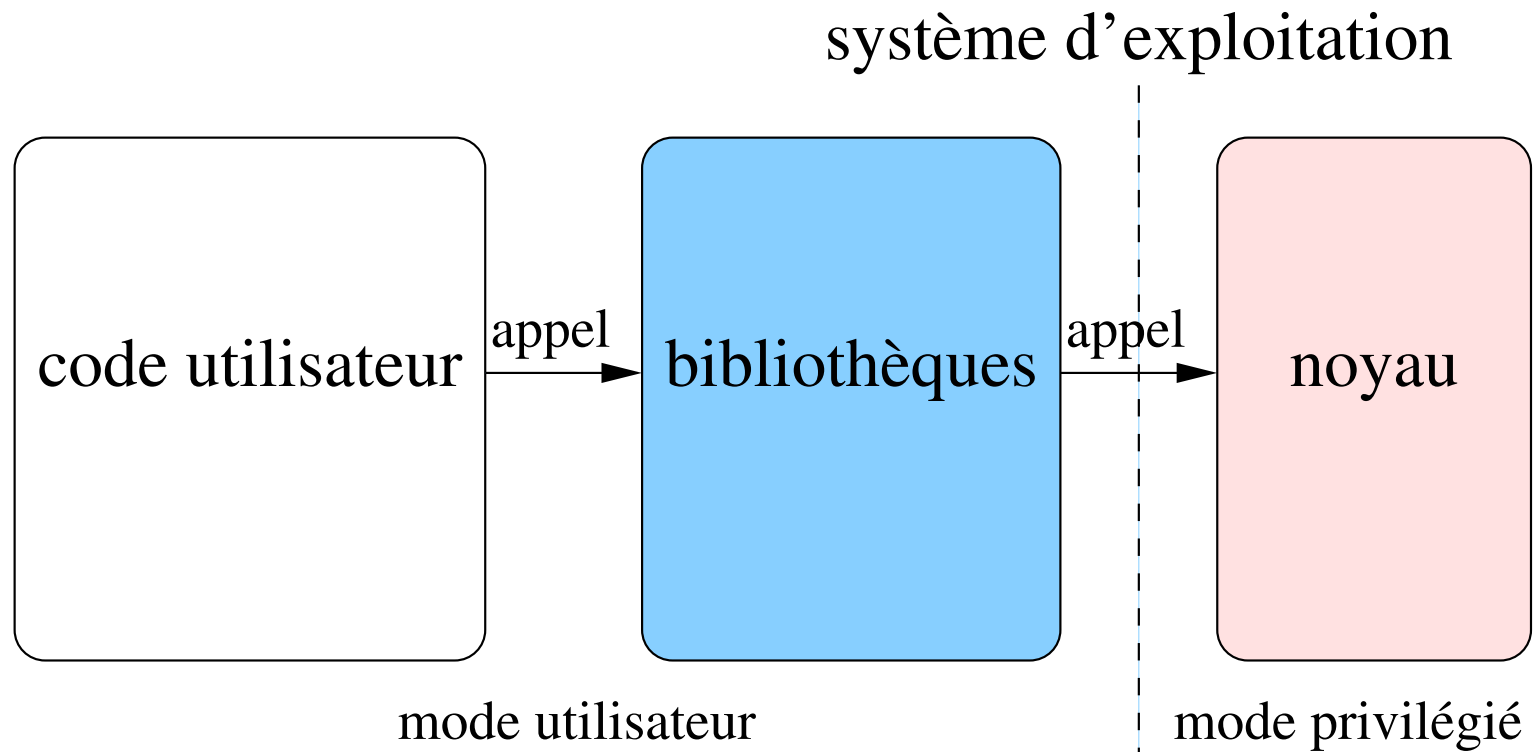
On distingue les composants suivants :

Le **noyau** (*kernel*): code privilégié qui gère l'interaction entre les processus et entre les périphériques

Les **bibliothèques** qui réalisent une interface standardisée entre applications et noyau

Applications de bas niveau (shell, ls, ...)

Fonctionnement des appels système:



Les bibliothèques système

Interface standardisé entre code utilisateur et noyau.

Dans Linux : (`hello.c`)

`ltrace` affiche les appels bibliothèque d'un processus

`strace` affiche les appels systèmes dans la forme des appels bibliothèque.

Pages man :

Section 1 : Applications de bas niveau

Section 2 : Bibliothèque système

Section 3 : Autres appels de bibliothèque (p.ex. `printf`)

etc

La norme POSIX

Pour permettre la compatibilité entre différents systèmes, les vendeurs ont créé des **normes**.

Dans ce cours on traite la norme **POSIX** qui est réalisée par plusieurs systèmes importants :

MacOS (complètement compatible)

Linux (largement compatible)

Windows (partiellement, p.ex. pour fichiers et réseaux)

Liens statiques et dynamiques

La liaison entre le code d'un programme et les bibliothèques peut être ...

... **dynamique** : (à défaut)

- code bibliothèque non inclus dans fichier exécutable ; bibliothèques partagées

- le fichier exécutable contient des informations sur les bibliothèques requises (`ldd`)

- lors du lancement, un *éditeur de liens* assure que tout appel pointe vers la bonne adresse

... **statique** :

- le code des appels bibliothèque est inclus dans un fichier exécutable lors de la compilation

- Drapeau `-static` dans gcc

Applications de bas niveau

Ensemble de programmes pour interagir avec le système.

Exemples: ligne de commande et d'autres programmes associés (`ls`, `cat`, ...)

On pourrait y ajouter l'interface graphique – la distinction entre programmes “du système” et “applications” n'est pas toujours précise.

Domaines d'un système d'exploitation

Processus, signaux, ordonnancement

Fichiers, réseau, entrées/sorties en général

Gestion de mémoire

Gestion d'utilisateurs

(éventuellement interface graphique)

...