**Structure d’un système informatique**

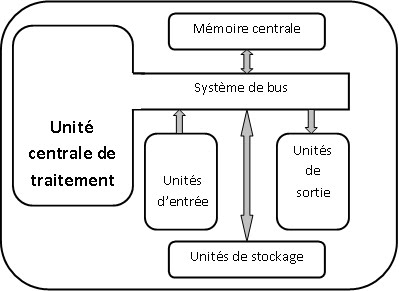
Introduction

Un système informatique peut être défini comme un ensemble de moyens matériels et logiciels nécessaire pour satisfaire les besoins en traitement, conservation et restitution des données des utilisateurs. Un ordinateur est une machine électronique ayant des composants matériels et logiciels.

**1 Structure d’un ordinateur**

La structure de base d’un ordinateur comprend les éléments fondamentaux suivants :

* Une unité centrale de traitement (unité centrale).
* Une unité de mémoire, appelée mémoire centrale.
* Des unités de stockage.
* Des unités d’entrée/sortie.
* Un système de bus permettant de véhiculer l’information entre l’unité centrale et les autres unités.



*Figure 1: Architecture d’un système informatique*

Les différents constituants de l’ordinateur sont composés de circuits électroniques ou d’éléments magnétiques. Ils fonctionnent donc avec des impulsions électriques ou font intervenir un champ magnétique, et donnent lieu à deux états (ouvert ou fermé) ou bien (nord ou sud).

L’information élémentaire est le ‘Bit’ un bit ne peut prendre que deux valeurs notées : 0 et 1.

**2 Unité centrale de traitement**

L’unité centrale de traitement ou CPU (Central Processing Unit) est le centre de calcul et de contrôle d’un ordinateur, elle constitue le « cerveau » de l’ordinateur.

L’unité centrale qui réalise toutes les opérations demandées, elle est matérialisée physiquement par un circuit électronique appelé « microprocesseur ». Un microprocesseur est caractérisé par :

* Sa marque (Intel ou cyrix).
* Sa fréquence d’horloge : le nombre d’opération que le microprocesseur peut effectuer en seconde (la fréquence atteint actuellement quelques Gigahertz).

1 KHz = 210 Hz

1 MHz = 220 Hz

1 GHz = 230 Hz

L’unité centrale comporte les éléments suivants :

* Unité arithmétique et logique : L’UAL est l’organe qui permet d’effectuer des opérations arithmétiques (addition, soustraction, multiplication, division), des opérations logiques (par exemple des comparaisons).
* Unité de commande : L’unité de commande gère le déroulement des instructions. Elle décode et exécute les instructions demandées.

**3 Mémoire centrale :**

La mémoire centrale est un organe de l’ordinateur permettant d’enregistrer, de stocker et de restituer les informations. La mémoire centrale se présente sous la forme de petites barrettes que l’on peut enficher dans des supports, appelés Slots. La mémoire centrale d’un ordinateur est séparée en deux sections : la mémoire vive (mémoire volatile destinée à être lue et écrite) et la mémoire morte (mémoire non volatile destinée à être lue).

**4 Les périphériques**

Les périphériques sont les organes de l’ordinateur qui permettent de communiquer avec l’unité centrale et de stocker les informations d’une façon permanente. On en distingue trois catégories à savoir : les périphériques d’entrée, les périphériques de sortie et les périphériques d’entrée-sortie.

**Évaluation**

Qu’est-ce qu’un ordinateur ?

Quels sont les principaux composants d’un ordinateur ?

Qu’est-ce qu’un périphérique ? Donnez les trois grandes catégories de périphériques

Qu’est-ce qu’un système informatique ?

A quoi sert un système d’exploitation

**SE comme gestionnaire de ressources**

Introduction

Le SE étant la partie qui fait fonctionner les composant d’un ordinateur, nous allons dans cette activité focaliser notre attention sur ce logiciel system. Qu’est-ce qu’un SE ?

**Définition**

Le système d’exploitation peut être défini comme programme système qui commande l’exécution de programmes d’application. Il est responsable pour le chargement et l’exécution de programmes d’application. Il doit veiller à la disponibilité des configurations matérielles et logicielles requises avant d’exécuter le programme d’application

**Rôle d’un SE**

Dans un système informatique, le SE est le programme qui agit comme intermédiaire entre l’utilisateur est la partie matérielle. Un système informatique possède un ensemble de ressources pour la circulation, le traitement et le stockage de données.

Ainsi, du point de vue machine, un SE peut être considéré comme gestionnaire de ressources.

Un ensemble de demande d’accès aux ressources, pouvant même générés des conflits sont gérés par le SE. C’est lui qui décide qui accède à telle ressource, pour telle durée en garantissant un fonctionnement efficace du système. Le SE contrôle également les dispositifs d’E/S et les logiciels d’application afin de garantir un bon fonctionnement du système.

**Conclusion**

Un SE est un logiciel indispensable pour le fonctionnement de l’ordinateur. Il joue le rôle d’intermédiaire entre la partie matérielle et les utilisateurs.

**Évaluation**

Distinguer les logiciels d’application des logiciels d’exploitation.

Qu’est-ce qui garantit la bonne utilisation des ressources sur un ordinateur ?

Existe-t-il un logiciel d’application indispensable au fonctionnement de l’ordinateur ? Justifier votre réponse.

**Evolution des SE**

Introduction

Les ordinateurs ont évolué dans le temps. Cette évolution a été largement influencée par l’évolution des SE. Mais quelles sont les évolutions qu’a connues le système d’exploitation ? Cette question sera répondue dans cette activité.

**Evolution**

Étudier l’évolution des systèmes d’exploitation est utilisé pour comprendre les exigences clés d’un système d’exploitation. De plus, elle aide à comprendre la signification des principales caractéristiques des systèmes d’exploitation modernes.

Les systèmes d’exploitation et l’architecture de l’ordinateur sont historiquement liés. La combinaison de l’architecture de l’ordinateur avec un système d’exploitation est connue sous le nom de plate-forme informatique. Ces changements architecturaux affectent la structure et la performance des systèmes d’exploitation.

Les premiers ordinateurs ne possédaient pas vraiment de système d’exploitation. Le moniteur était chargé avec l’unique programme à exécuter qui se greffait sur ce système rudimentaire tout comme une fonction ou un sous-programme se greffe aujourd’hui sur le programme principal. Les entrées-sorties étaient rudimentaires, réduites à un ruban perforé, les programmes écrits en langage machine, c’est à dire directement codés en suite de chiffres, exécutés pas à pas et modifiés directement en mémoire au moyen d’un jeu de clés !

Vers 1950 est apparu le moniteur d’enchaînement. Il permettait l’exécution d’une série de travaux en séquence, c’est à dire l’un après l’autre. C’est l’ancêtre du traitement par lot. Il possédait déjà des fonctions de protection: maximum de temps par travail, supervision des périphériques, protection de la zone mémoire utilisée, gestion et abandon des travaux erronés. Ces avancées étaient fondamentales car elles introduisaient un environnement autour du programme à exécuter.

L’informatique moderne naît dans les années 1960. On peut résumer rapidement ses avancées autour de l’invention des notions suivantes :

* Apparition des processeurs d’entrées-sorties
* Multiprogrammation, c’est à dire possibilité d’exécuter plusieurs programmes simultanément.
* Compilateurs
* Temps partagé
* Mémoire paginée virtuelle. Elle permet de faire fonctionner un ensemble de programmes dont la taille est supérieure à celle de la mémoire physique.
* Les communications

**Évaluation**

Donner les grandes étapes de l’évolution des SE. Expliquer brièvement les caractéristiques de chaque étape.

Comment l’évolution des SE a-t-elle influencé celle des ordinateurs ?

**Structure d’un SE**

Introduction

Dans cette activité nous allons détailler la structure interne d’un système d’exploitation. Plusieurs approches d’implémentation des SE existe, nous allons donner un aperçu sur quelques-uns. Il s’agit notamment des systèmes monolithiques, des systèmes en couche, des machines virtuelles, des exonoyaux et des systèmes client-serveur.

**Système monolithique**

Le SE est une structure hiérarchique écrit comme un ensemble de procédures dont chacune peut appeler n’importe quelle autre en cas de besoin. La structure de base suggérée par cette organisation est la suivante :

* un programme principal qui invoque la procédure de service ;
* un ensemble de procédures de services qui gèrent les appels système ;
* un ensemble de procédures utilitaires auxiliaires des procédures.

**Les systèmes en couche**

L’organisation d’un système en une série de couches est une généralisation de la structure précédente. Le système était composé par une série de couches hiérarchiques.

* La couche 0 fournissait le service de multiprogrammation CPU.
* La couche 1 était en charge de la gestion de la mémoire et des tampons
* La couche 2 se chargeait de la communication interprocessus et console
* La couche 3 était en charge des entrées sorties
* La couche 4 contenait les programmes utilisateurs
* La couche 5 était le processus opérateur.

**Machines virtuelles**

La plus part du temps, les utilisateurs des ordinateurs sont assis devant leur PC (unité centrale

* les différents périphériques). Cependant, dans d’autres cas, les utilisateurs sont en face d’un terminal connecté à un mainframe ou ordinateur connecté à d’autres ordinateurs en réseau.

Dans le premier cas, le système est conçu de telle sorte qu’un seul utilisateur monopolise toutes les ressources. Ainsi l’objectif de tels systèmes est l’efficacité dans l’accomplissement des tâches soumises par l’utilisateur. Ces systèmes sont donc conçus de telle sorte qu’elles soient faciles à utiliser avec pour seule considération les performances du système, sans se soucier de l’utilisation des ressources. Dans le second cas (utilisation d’un terminal), la même ressource peut être demandée par plusieurs utilisateurs chacun assis devant son propre terminal.

Dans de tels cas, la conception de ces SE met un accès particulier sur l’accès aux ressources. Ils doivent garantir la disponibilité de ressources et un partage équitable entre les requérants.

En dernier lieu, on distingue de systèmes où l’utilisateur possède des ressources dédiées (utilisation d’un ordinateur personnel) mais comportant des ressources partagées (serveur de fichiers, serveur d’impression, etc.).

Dans ce cas, la conception de tels SE doit chercher un compromis entre la gestion des ressources et l’utilisation individuelle de la machine.

L’ensemble de ces situations et différentes exigences des utilisateurs, nous montre qu’un SE est un système conçu de telle sorte qu’il garantit une facilité d’utilisation, une disponibilité et un partage des ressources dans un environnement où les utilisateurs ont besoins et exigences différents.

**Les exonoyaux**

C’est un système qui donne à chaque utilisateur un clone de l’ordinateur réel mais seulement pourvu d’un sous ensemble de ses ressources. L’exonoyau est la couche la plus basse ayant pour rôle d’allouer des ressources pour les machines virtuelles et de vérifier les tentatives d’accès à ces dernières.

**Le modèle client-serveur**

C’est la tendance actuelle des SE. Elle consiste à implémenter la plus grande partie du SE sous forme de processus utilisateur. La demande d’un service passe par l’envoie du service demandeur (le client) d’une requête au service pourvoyeur (serveur) qui rend le service et renvoie le résultat.

**Conclusion**

Les différents modèles des SE reposent sur une architecture bien définie. La première approche était la structure monolithique. Cette approche a été améliorée passant de la structure en couche, a été considérablement améliorée pour arriver maintenant à la structure actuelle qui est le système client-serveur.

**Évaluation**

* Indiquer les différents modèles des structures interne des SE
* Pourquoi la structure monolithique était une sorte de gâchis ?
* Quelles sont les spécificités du modèle client-serveur ?

# Parallélisme ou multitâche ?

En simplifiant (un peu) moins, [ce qui compose un processeur](https://fr.wikipedia.org/wiki/Processeur) :

* [UAL, Unité arithmétique et logique](https://fr.wikipedia.org/wiki/Unit%C3%A9_arithm%C3%A9tique_et_logique) : prend en charge les calculs arithmétiques élémentaires et les tests.
* [Unité de contrôle, ou séquenceur](https://fr.wikipedia.org/wiki/Unit%C3%A9_de_contr%C3%B4le) : permet de synchroniser les différents éléments du processeur.
* [Compteur ordinal](https://fr.wikipedia.org/wiki/Compteur_ordinal) : contient l’adresse mémoire de l’instruction en cours d’exécution ou de la suivante, selon l'architecture.
* Registre d’instructions : contient l’instruction en cours de traitement.
* L’horloge qui synchronise toutes les actions de l’unité centrale.
* Trois types de bus :
  + un bus de données, qui définit la taille des données pour les entrées/sorties, dont les accès à la mémoire ;
  + un bus d'adresse, qui permet, lors d'une lecture ou une écriture, d'envoyer l'adresse où elle s'effectue, et donc définit le nombre de cases mémoire accessibles ;
  + un bus de contrôle permet la gestion du matériel, via les interruptions.

# Multitâche

Le fonctionnement interne d'un ordinateur est fondamentalement séquentiel, contrôlé par le séquenceur. Mais plusieurs méthodes existent pour (nous donner l'illusion) que plusieurs choses se passent en même temps. La première en date a été le multitâche ([Gamma 60](https://fr.wikipedia.org/wiki/Gamma_60), de Bull, en 1958).

Un système d'exploitation est multitâche s’il permet d’exécuter, de façon apparemment simultanée, plusieurs programmes informatiques.

La simultanéité apparente est le résultat de l’alternance rapide d’exécution des processus présents en mémoire.

Le passage de l’exécution d’un processus à un autre est appelé commutation de contexte. Ces commutations peuvent être initiées par les programmes eux-mêmes (multitâche coopératif) ou par le système d’exploitation lors d’événements externes (multitâche préemptif).

Les systèmes multitâches se sont ensuite répandus dans le courant des années 1960 dans une double déclinaison :

* systèmes à temps partagé : ils permettent à plusieurs utilisateurs de travailler sur la même machine (pour des questions de rentabilité, le matériel étant extrêmement coûteux) ;
* systèmes temps réel pour la commande de processus industriels : de nombreux programmes de petite taille s'exécutent à la réception d'événements externes (acquisition de données, alerte d'un équipement...) dans un mode de gestion par interruptions avec priorités d'exécution.

Bien que, au départ, le multitâche ait été conçu pour permettre à plusieurs utilisateurs d’utiliser le même ordinateur, il est apparu très rapidement qu’il était très pratique même pour un seul utilisateur. Utiliser son traitement de texte favori tout en naviguant sur le Web est de nos jours une caractéristique incontournable.

Une autre utilité du multitâche provient du fait qu’il est plus facile de concevoir et d’écrire plusieurs programmes simples, plutôt qu’un seul programme capable de tout faire, puis de les faire coopérer pour effectuer les tâches nécessaires.

Les algorithmes implémentant le multitâche ont été raffinés au cours du temps. Les systèmes modernes peuvent gérer des processus avec des priorités différentes ainsi que des ordinateurs comportant de un à quelques centaines de processeurs.

# Multitâche coopératif

Le multitâche coopératif est une forme simple de multitâche où chaque processus doit explicitement permettre à une autre tâche de s’exécuter. Il a été utilisé, par exemple, dans les produits Microsoft Windows jusqu’à Windows 3.11 ou dans Mac OS jusqu’à Mac OS 9. Cette approche simplifie l’architecture du système d’exploitation mais présente plusieurs inconvénients :

* Si un des processus ne redonne pas la main à un autre processus, par exemple si le processus est buggé, le système entier peut s’arrêter.
* Le partage des ressources (temps CPU, mémoire, accès disque, etc.) peut être inefficace.
* Le multitâche coopératif est une forme de couplage fort.

# Multitâche préemptif

Pour remédier à cette situation, les systèmes grand public ont évolué pour utiliser une approche nommée « multitâche préemptif ».

Dans un tel système, le processeur signale au système d’exploitation que le processus en cours d’exécution doit être mis en pause pour permettre l’exécution d’un autre processus. Le système doit alors sauver l’état du processus en cours (le compteur ordinal, les valeurs des registres) ; le processus est placé dans une file d’attente ; lorsqu’il est en bout de file, son contexte d’exécution est restauré. À la différence du « multitâche coopératif », du point de vue des programmes préemptés, la commutation de contexte est transparente.

Unix (1969) et ses dérivés (BSD en 1977, Linux en 1991, ...) sont des systèmes basés sur le multitâche préemptif.

Sinclair QDOS (1984), AmigaOS (1985), NeXTSTEP (1989) ou BeOS (1991) sont des systèmes multitâches préemptif depuis leur naissance.

D’autres systèmes d’exploitation le sont devenus comme Windows (avec Windows NT 3.1 en 1993 pour les professionnels et Windows 95 en 1995 pour le grand public) et MacOS X en 2001.

# Préemptif ou coopératif ?

Le multitâche préemptif est plus robuste que le multitâche coopératif : une tâche ne peut bloquer l’ensemble du système. Le système d’exploitation peut aussi utiliser plus efficacement les ressources disponibles, par exemple si des données sont disponibles via un périphérique d’entrée, le processus devant traiter ces données peut être immédiatement activé.

De plus, une tâche en attente de données ne consommera pas de temps processeur avant que ses données ne soient réellement disponibles.

# Parallélisme

Le parallélisme consiste à mettre en œuvre des architectures permettant de traiter des informations de manière simultanée. Ces techniques ont pour but de réaliser le plus grand nombre d'opérations en un temps le plus petit possible.

Les architectures parallèles sont devenues le paradigme dominant pour tous les ordinateurs depuis les années 2000. En effet, la vitesse de traitement qui est liée à l'augmentation de la fréquence des processeurs connait des limites. La création de processeurs multi-cœurs, traitant plusieurs instructions en même temps au sein du même composant, résout ce dilemme pour les machines de bureau depuis le milieu des années 2000.

Certains types de calculs se prêtent particulièrement bien à la parallélisation : la dynamique des fluides, les prédictions météorologique, la modélisation et simulation de problèmes de dimensions plus grandes, le traitement de l'information et l'exploration de données, le décryptage de messages, la recherche de mots de passe, le traitement d'images ou la fabrication d'images de synthèse, tels que le lancer de rayon, l'intelligence artificielle et la fabrication automatisée.

Mais tout ne peut pas être parallélisé. Prenons par exemple ces deux fonctions :

|  |  |
| --- | --- |
| **function Dep(a, b) :   c = a \* b   d = 2 \* c** | **function NoDep(a, b) :   c = a \* b   d = 2 \* b   e = a + b** |

La première fonction n'est clairement pas parallélisable, car sa deuxième instruction dépend du résultat de la première et devra attendre que celle-ci soit terminée pour s'exécuter.

La deuxième fonction est par contre bien parallélisable, car ses trois instructions sont indépendantes les unes des autres.

# Quel parallélisme ?

On peut distinguer trois principaux types de parallélisme : les pipelines, les ordinateurs multi-processeurs, et les processeurs multi-cœurs.

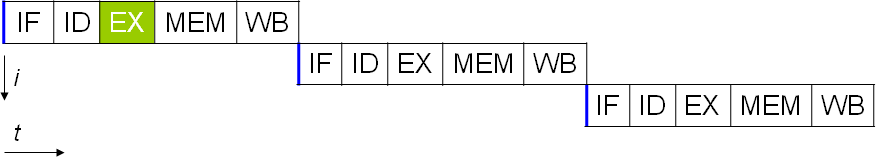
# Pipelines

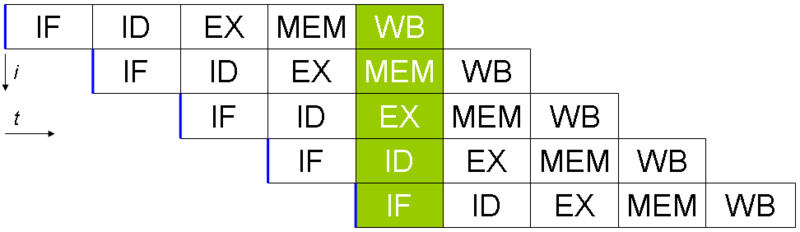
Un programme informatique est, par essence, un flux d'instructions exécuté par un processeur. Chaque instruction nécessite plusieurs cycles d'horloge, l'instruction est exécutée en autant d'étapes que de cycles nécessaires :

* IF (Instruction Fetch) charge l'instruction à exécuter dans le pipeline.
* ID (Instruction Decode) décode l'instruction et adresse les registres.
* EX (Execute) exécute l'instruction (par la ou les unités arithmétiques et logiques).
* MEM (Memory), dénote un transfert depuis un registre vers la mémoire dans le cas d'une instruction du type STORE (accès en écriture) et de la mémoire vers un registre dans le cas d'un LOAD (accès en lecture).
* WB (Write Back) stocke le résultat dans un registre. La source peut être la mémoire ou bien un registre.

Les microprocesseurs séquentiels exécutent l'instruction suivante lorsqu'ils ont terminé la première.

En supposant que chaque étape met 1 cycle d'horloge pour s'exécuter, il faut normalement 5 cycles pour exécuter une instruction, 15 pour 3 instructions :

  
Séquençage des instructions dans un processeur sans pipeline. Il faut 15 cycles pour exécuter 3 instructions.Dans le cas du parallélisme d'instruction, le microprocesseur peut traiter plusieurs de ces étapes en même temps pour plusieurs instructions différentes, car elles ne mobilisent pas les mêmes ressources internes. Autrement dit, le processeur exécute en parallèle des instructions qui se suivent à différents stades d'achèvement. Cette file d'exécution s'appelle un pipeline. Ce mécanisme a été mis en œuvre pour la première fois dans les années 1960 par IBM.

  
Séquençage des instructions dans un processeur doté d'un pipeline à 5 étages.  
Il faut 9 cycles pour exécuter 5 instructions.  
À t = 5, tous les étages du pipeline sont sollicités, et les 5 opérations ont lieu en même temps.

# Multi-processeurs

L'idée de faire cohabiter deux processeurs dans la même machine date des années 1960, le D825 de Burroughs Corporation commercialisé en 1962 est le premier ordinateur multi-processeur, mais ce système n'était pas parallèle. Il a fallu attendre 1969 pour que Honeywell produise le premier ordinateur qui dispose de processeurs fonctionnant réellement en parallèle. Les huit processeurs de cette machine de la série Honeywell 800 fonctionnaient de manière symétrique (ou SMP), c'est-à-dire que tous les processeurs ont la même fonction et les mêmes capacités. Ce ne fut pas le cas de toutes les machines, DEC et le MIT ont développé dans les années 1970 une technologie asymétrique, mais elle a été abandonnée dans les années 1980. Peu de machines ont utilisé ce principe, même si certaines ont eu du succès comme les VAX.

# Multi-cœurs

Un microprocesseur multi-cœur (multi-core en anglais) est un processeur possédant plusieurs cœurs physiques fonctionnant simultanément. Il se distingue d'architectures plus anciennes où un processeur unique commandait plusieurs circuits de calcul simultanés.

Un cœur physique est un ensemble de circuits capables d’exécuter des programmes de façon autonome. Toutes les fonctionnalités nécessaires à l’exécution d'un programme sont présentes dans ces cœurs : compteur ordinal, registres, unités de calcul, etc. Des caches sont définis pour chaque processeur ou partagés entre eux.

# Cas particuliers

* Coprocesseurs

Les coprocesseurs sont des unités destinées à fournir des fonctionnalités supplémentaires au processeur, ou des circuits spécialisés pour des applications particulières. Certaines de ces applications sont facilement parallélisables, comme les calculs graphiques. Beaucoup de coprocesseurs graphiques actuels permettent aussi de réaliser des calculs parallèles sans qu'il s'agisse nécessairement de création d'image.

* Systèmes à plusieurs machines

Pour faire fonctionner en parallèle un plus grand nombre de processeurs, on peut utiliser plusieurs machines qui communiquent ensemble via un réseau. Selon les cas, il peut s'agir d'un réseau d'interconnexion spécialement conçu ou d'internet.

Système embarqué

# Introduction

Un **système embarqué** peut être défini comme un système électronique et informatique autonome, qui est dédié à une tâche bien précise. Ses ressources disponibles sont généralement limitées. Cette limitation est généralement d'ordre spatial (taille limitée) et énergétique (consommation restreinte).

Les systèmes embarqués font très souvent appel à l'[informatique](https://www.techno-science.net/glossaire-definition/Informatique.html), et notamment aux systèmes [temps](https://www.techno-science.net/glossaire-definition/Temps.html) réel.

Le terme de [système embarqué](https://www.techno-science.net/glossaire-definition/Systeme-embarque.html) désigne aussi bien le matériel que le [logiciel](https://www.techno-science.net/glossaire-definition/Logiciel.html) utilisé.

Un synonyme de système embarqué est système enfoui.

# Histoire

Le premier système moderne embarqué reconnaissable a été le Apollo Guidance Computer, le système de guidage de la mission [lunaire](https://www.techno-science.net/glossaire-definition/Lunaire.html) Apollo, développé par Charles Stark Draper du Massachusetts Institute of Technology.

# Architecture

Les systèmes embarqués utilisent généralement des microprocesseurs à basse consommation d'[énergie](https://www.techno-science.net/definition/1724.html) ou des microcontrôleurs, dont la partie logicielle est en partie ou entièrement programmée dans le matériel, généralement en [mémoire](https://www.techno-science.net/definition/10848.html) dans une mémoire morte (ROM), EPROM, EEPROM, FLASH, etc. (on parle alors de firmware).

# Contraintes

Les systèmes embarqués exécutent des tâches prédéfinies et ont un [cahier des charges](https://www.techno-science.net/definition/690.html) contraignant à remplir, qui peut être d'ordre :

* De coût. Le prix de revient doit être le plus faible possible surtout s'il est produit en grande série.
* D'espace compté, ayant un espace mémoire limité.
* De [puissance](https://www.techno-science.net/definition/1737.html) de calcul. Il convient d'avoir la puissance de calcul juste nécessaire pour répondre aux besoins et au contraintes temporelles de la tâche prédéfinie.
* Temporel, dont les temps d'exécution et l'échéance temporelle d’une tâche sont déterminés (les délais sont connus ou bornés a priori).
* De sûreté de fonctionnement. Car s'il arrive que certains de ces systèmes embarqués subissent une défaillance, ils mettent des vies humaines en danger ou mettent en périls des investissements importants.
* De sécurité. Ces systèmes peuvent se révéler être porteurs d'informations confidentielles pour leur(s) utilisateur(s), qu'il convient de conserver et de protéger. Notamment, en ce qui concerne l'[acquisition](https://www.techno-science.net/definition/2673.html) et la transmission d'informations médicales.

# Domaines d'applications

Les domaines dans lesquels on trouve des systèmes embarqués sont de plus en plus nombreux :

* transport : Automobile, Aéronautique (avionique), etc.
* astronautique : fusée, [satellite artificiel](https://www.techno-science.net/definition/2668.html), [sonde spatiale](https://www.techno-science.net/glossaire-definition/Sonde-spatiale.html), etc.
* militaire : [missile](https://www.techno-science.net/glossaire-definition/Missile.html)
* télécommunication : [Set-top box](https://www.techno-science.net/definition/11578.html), [téléphonie](https://www.techno-science.net/definition/10721.html), [routeur](https://www.techno-science.net/glossaire-definition/Routeur.html), pare-feu, serveur de [temps](https://www.techno-science.net/glossaire-definition/Temps.html), [téléphone](https://www.techno-science.net/definition/3986.html) portable, etc.
* électroménager : [télévision](https://www.techno-science.net/definition/10722.html), [four](https://www.techno-science.net/glossaire-definition/Four.html) à micro-ondes
* impression : [imprimante multifonctions](https://www.techno-science.net/definition/11223.html), [photocopieur](https://www.techno-science.net/definition/8121.html), etc.
* informatique : [disque dur](https://www.techno-science.net/glossaire-definition/Disque-dur.html), [Lecteur de disquette](https://www.techno-science.net/glossaire-definition/Lecteur-de-disquette.html), etc.
* multimédia : console de jeux [vidéo](https://www.techno-science.net/definition/7376.html), [assistant personnel](https://www.techno-science.net/glossaire-definition/Assistant-personnel.html)
* guichet [automatique](https://www.techno-science.net/definition/5736.html) bancaire (GAB)
* équipement médical
* [automate programmable industriel](https://www.techno-science.net/definition/6694.html), contrôle-commande
* jeux : consoles
* [métrologie](https://www.techno-science.net/definition/1684.html)

# Caractéristiques

* Plutôt que des systèmes universels effectuant plusieurs tâches, les systèmes embarqués sont étudiés pour effectuer des tâches précises. Certains doivent répondre à des contraintes de temps réel pour des raisons de [fiabilité](https://www.techno-science.net/glossaire-definition/Fiabilite.html) et de rentabilité. D'autres ayant peu de contraintes au niveau performances permettent de simplifier le système et de réduire les couts de fabrication.
* Les systèmes embarqués ne sont pas toujours des modules indépendants. Le plus souvent ils sont intégrés dans le dispositif qu'ils contrôlent.
* Le [logiciel](https://www.techno-science.net/glossaire-definition/Logiciel.html) créé pour les systèmes embarqués est appelé [firmware](https://www.techno-science.net/definition/655.html). Il est stocké dans de la [mémoire](https://www.techno-science.net/definition/10848.html) en lecture seule ou de la [mémoire flash](https://www.techno-science.net/glossaire-definition/Memoire-flash.html) plutôt que dans un [disque](https://www.techno-science.net/definition/3594.html) dur. Il fonctionne le plus souvent avec des ressources matérielles limitées : un petit voire pas de clavier, un petit [écran](https://www.techno-science.net/definition/8155.html) et peu de mémoire.

# [Interface](https://www.techno-science.net/definition/3769.html) utilisateur

Certains systèmes embarqués peuvent ne pas avoir d'interface utilisateur (ils sont alors dédiés à une seule tâche). Mais cette interface peut également être similaire à celle d'un système d'exploitation d'[ordinateur](https://www.techno-science.net/glossaire-definition/Ordinateur.html) (par exemple un PDA).

* Les systèmes les plus simples comportent uniquement des boutons, des LED.
* Les systèmes les plus complexes peuvent avoir un [écran tactile](https://www.techno-science.net/definition/8136.html) ou encore un écran comportant des boutons de façon à minimiser l'espace. La signification des boutons change selon l'écran et la sélection se fait naturellement en pointant la fonction désirée.
* Les ordinateurs de poche possèdent en général un bouton style "joystick" pour la [navigation](https://www.techno-science.net/glossaire-definition/Navigation.html).
* Avec l'[explosion](https://www.techno-science.net/glossaire-definition/Explosion.html) du web, les fabricants de systèmes embarqués ont proposé une nouvelle option : une interface style [page web](https://www.techno-science.net/glossaire-definition/Page-Web.html) sur une connexion au [réseau](https://www.techno-science.net/definition/3799.html). Cela permet d'éviter le coût d'un système sophistiqué [tout](https://www.techno-science.net/glossaire-definition/Tout.html) en conservant une interface complète sur un autre ordinateur, quand cela est nécessaire. Interface couronné de succès pour les installations permanentes à distance, les routeurs en particulier.

# Fiabilité

Les systèmes embarqués sont la plupart du temps dans des machines qui doivent fonctionner en continu pendant de nombreuses années, sans erreurs et, dans certains cas, réparer eux-mêmes les erreurs quand elles arrivent. C'est pourquoi les logiciels sont toujours développés et testés avec plus d'attention que ceux pour les PC. Les pièces mobiles non fiables (par exemple les lecteurs de disques, boutons ou commutateurs) sont proscrites.

La question de la fiabilité peut inclure :

* Le système ne peut pas être éteint pour des réparations ou ce sont des réparations inaccessibles.

La solution peut être des pièces détachées supplémentaires ou un "mode mou" du logiciel qui fournit un fonctionnement [partiel](https://www.techno-science.net/definition/1268.html).

Par exemple : les câbles sous-marins, les balises de navigation, les puits de forage...

* Le système doit rester en [marche](https://www.techno-science.net/glossaire-definition/Marche.html) pour des raisons de sécurité. Souvent, les sauvegardes sont effectuées par un [opérateur](https://www.techno-science.net/definition/5057.html).

Dans ce cas, le « mode mou » est toléré.

Par exemple : les systèmes de [contrôle](https://www.techno-science.net/definition/2787.html) des réacteurs, les usines chimiques, les signaux de [train](https://www.techno-science.net/glossaire-definition/Train.html)...

* Un arrêt du système peut provoquer des pertes monétaires énormes s'il s'éteint.

Par exemple : les systèmes de ponts ou d'ascenseurs, les transferts de fond, les salles de bourse, les ventes ou services automatiques...

**Système temps réel**

Introduction

Un système d’exploitation en temps réel est la catégorie d’un système d’exploitation qui est utilisé pour gérer des applications en temps réel. C’est-à-dire que pour de tel système l’on a besoin d’une réponse correcte et dans un temps très court. Ce genre de système possède des spécificités autres que celles déjà vues dans les unités précédentes. Cette activité consiste en un aperçu sur les systèmes temps réels.

**Exigences des systèmes temps réels**

Un système en temps réel est généralement un système à but spécifique, de petite taille, à peu de frais produites en masse et avec les exigences de calendrier précis. Un système temps réel est conçu pour un seul but et vous ne pouvez pas l’utiliser pour d’autres tâches que celle spécifiée lors de sa conception. Les systèmes temps réels autonome, conçus pour des problèmes spécifiques, pour lesquels le matériel et l’application sont intimement liés sont appelés des systèmes embarqués.

La plupart des systèmes en temps réel sont utilisés dans des environnements exigus qui place également une limitation de CPU et de mémoire pour ces systèmes. Comme ces systèmes sont aussi le plus souvent utilisés avec les appareils ménagers et les appareils grand public qui sont fabriqués en masse dans des environnements de coûts conscients, les processeurs de ces systèmes doivent également être fabriqués par un coût moindre. Enfin, l’exigence de temps une caractéristique qui définit un système en temps réel qui permet de réagir aux événements en temps opportun. En général, un système en temps réel a trois exigences principales :

1. Prévisibilité: le comportement de synchronisation du système d’exploitation doit être prévisible
   1. Gestion: le système d’exploitation doit gérer le calendrier et la programmation de car il a peut-être au courant des délais de tâches. En outre, le système d’exploitation doit fournir des services de temps précis avec une résolution élevée.
   2. Vitesse: Le système d’exploitation doit être rapide dans toutes les réponses, ce qui est important dans la pratique.

Les systèmes d’exploitation traditionnels ont la capacité de supporter plusieurs périphériques comme entrée, sortie, etc., plusieurs utilisateurs, des mécanismes de protection contre toute menace possible, et la virtualisation pour améliorer son espace mémoire où tous ces éléments créent une certaine complexité sur le noyau du système. En revanche, les systèmes temps réel sont conçus simplement avec un petit nombre de codes et ont aucune de ces fonctionnalités. Les principales raisons pour lesquelles les systèmes en temps réel ne parviennent pas à le faire sont les suivantes:

* La plupart des systèmes en temps réel sont système conçu pour servir une seule tâche. De tel système ne connaissent pas la notion d’utilisateur. Le système prend en charge tout simplement un petit nombre de tâches, qui obtient souvent son entrée à partir de périphériques tels que des capteurs.
* Il est impossible de donner aux systèmes d’exploitation temps réels, les fonctions prises en charge par les systèmes d’exploitation de bureau standard sans processeurs rapides et des quantités abondantes de la mémoire qui sont indisponibles dans les systèmes en temps réel.
* Le soutien des caractéristiques communes dans les systèmes de bureau standard par les systèmes en temps réel les rendrait coûteux et économiquement impraticable.
* Les domaines d’application des systèmes temps réels sont de plus en plus variés, on en trouve dans : l’électroménager, transport, télécommunication, guichet automatique, etc.
* **Évaluation**
* Définir un système un système temps réel.
* Différencier un système temps réels d’un système classique
* Quel est l’importance des systèmes temps réel.
* Expliquer les trois exigences des systèmes temps réels.

**Sécurité et protection**

**Introduction**

Dans les entreprises, la sécurité des données est nécessaire pour qu’elles ne soient pas dérobées. Dès lors que de plus en plus de données soient stockées dans des systèmes informatiques, il est impératif qu’elles soient protégées. La protection des données contre une utilisation non autorisée est donc devenu un impératif pour tous les SE. Dans cette unité d’apprentissage, nous allons explorer les différentes attaques exercées sur la sécurité du système ainsi que des principes de conception à prendre en considération, tout en développant un système d’exploitation de sorte qu’un système peut avoir une résistance à ces types d’attaque.

**Sécurité informatique:** La sécurité des systèmes d’information estl’ensemble des moyens techniques, organisationnels, juridiques et humains nécessaire et mis en place pour conserver, rétablir, et garantir la sécurité du système d’information.

**Menace informatique:** Les menaces contre la sécurité informatiqueregroupent les actions susceptibles de nuire et de porter atteinte, partiellement ou totalement, à un système informatique. Dans ce contexte, les menaces peuvent être soustraites du cadre purement virtuel, propre au système numérique, et ainsi venir de l’extérieur ou de l’intervention humaine

**Attaque:** Une attaque est une exploitation d’une faille d’un système

Authentification: L’authentification l’identification d’un utilisateur afin de lui donner accès aux fonctionnalités d’un système.

**Virus:** Un virus informatique est un programme, généralement de

petite ou très petite taille, doté des propriétés suivantes : infection ; multiplication ; possession d’une fonction nocive. C’est un programme malveillant.

**Antivirus:** Logiciel conçu dans le but de protéger un ordinateur en luttantcontre les virus

**Faille:** Faiblesse d’un système informatique susceptible d’être exploitéeafin de porter atteinte à un système informatique

**Hacking :** Le hacking est un ensemble de techniques informatiques,visant à attaquer un réseau, un site, etc.

**Aperçu général sur la sécurité**

Introduction

Les ordinateurs les plus sécurisés sont ceux qui ne sont pas connectés à l’Internet. Dès lors que l’ordinateur est connecté sur Internet c’est une brèche à de nombreuses attaques. Alors devrons-nous abandonner de connecter nos ordinateurs à l’Internet ? La réponse est certes non. Alors comment peut-on travailler en mode connecté et garantir la sécurité des données et des applications ? Quelles sont les principales attaques dont on est le plus souvent cibles ? L’ensemble de ces question sont traités dans cette activité ?

**Concepts de sécurité**

Les principaux concepts auxquels s’appuie la sécurité des données sont :

**L’authentification :** Tout utilisateur devrait avoir des informations d’authentification quil’identifient de manière unique. Avec celles-ci le système devrait connaître son profil et finalement déterminer à quoi il a droit.

**La confidentialité :** C’est garantir que l’accès aux données est fait par l’utilisateur concerné.

Une des techniques qui permettent de de garantir la confidentialité est la cryptographie.

**Intégrité :** C’est le fait de protéger les données contre des modifications malveillantes.

**Disponibilité :** C’est garantir le fonctionnement. Eviter un déni de service de toute sorte

Le mécanisme de protection à mettre en place doit tenir en compte des menaces qu’encoure le système. Par menace on désigne tout évènement potentiel susceptible de provoquer un effet indésirable sur le système. Pour les données ça concerne essentiellement la modification, l’ajout, la suppression, la copie par des personnes/programmes qui ne le devraient pas.

**L’intrusion :** L’intrusion concerne l’usurpation d’identité afin d’accéder aux ressourcesauxquelles on n’a pas le droit.

**La perte accidentelle de données :** Un autre challenge concerne la perte accidentelledes données. Les conséquences de la perte accidentelle des données sont souvent catastrophiques essentiellement pour les données sensibles. La perte accidentelle des données est souvent due aux phénomènes naturels, les erreurs matérielles et systèmes ou les erreurs humaines.

**Évaluation**

1. Qu’est-ce qu’une menace ?
   1. Pourquoi attaque-t-on un système ?

**Les attaques du système**

Introduction

L’activité précédente a été une occasion de passer en revue les principales menaces et failles d’un système. Dans cette activité, nous allons voir les principales attaques dont fait l’objet le système. Un système peut être attaqué par des situations différentes agressions qui testent sa vulnérabilité. Les attaques peuvent être d’ordre physique qui consiste à voler ou physiquement les ressources du système d’endommager ou de logique qui consiste à utiliser les programmes malveillants et les ressources des systèmes de dégâts. Les attaques peuvent venir de l’intérieur comme de l’extérieur.

**Les attaques internes**

Ce genre d’attaques est effectué par une entité qui a un droit d’accès. L’attaquant peut voler ou endommager physiquement les ressources ou après pénétration réussie à un système, l’intrus peut effectuer son opération souhaitée et attaquer le système. Plusieurs attaques logiques à l’intérieur peuvent être imposées sur les systèmes. Ci-dessous, nous présentons quelques-unes d’entre elles:

Les chevaux de Troie: C’est un programme apparemment innocent, mais avec des codes à l’intérieur qui provoque un fonctionnement inattendu comme la modification, la suppression ou même transférer les informations de l’utilisateur sur l’emplacement de l’attaquant.

Le spoofing: l’attaquant présente une interface de connexion de faux qui ressemble exactement à la vraie interface de connexion. L’utilisateur donne alors son / ses données d’authentification de l’interface de faux qui recueille et transmet les informations à l’attaquant et les sorties. L’interface de connexion d’origine est affiché alors et l’utilisateur croit de faire une erreur précédemment sur l’information et pense que le système affiche l’interface pour la deuxième fois.

Bombe logique: C’est un code embarqué dans un programme qui “explose” lorsque certaines conditions sont remplies, par exemple une certaine date ou la présence de certains fichiers ou des utilisateurs.

Trappes: C’est un point d’entrée secrète dans un programme, souvent laissé par les développeurs du programme, ou parfois livré via une mise à jour logicielle. Ce sont des codes inclus dans un programme pour aider le contournement de l’attaquant des contrôles normaux.

Dépassement de mémoire tampon: C’est une attaque qui se passe sur un système en exploitant des bugs dans un programme et l’utilisation de ces bugs à endommager le système. Le bug peut être un simple cas de mauvaise programmation, dans lequel le programmeur a négligé de codes associée vérification

**Les attaques externes**

Ce sont des attaques provoquées par un attaquant de l’extérieur du système principalement sur un réseau. Ces attaques utilisent des codes transmis à la machine ciblée sur un réseau et provoque des dommages à la machines cible. Ces codes sont connus comme des codes malveillants et peuvent être virus, un ver, codes mobiles ou des applets Java.

Le virus est un programme qui se réplique en se fixant avec un autre programme et causant des lésions sur les ressources du système. L’attaquant distribue le virus en l’attachant à un programme attrayant qui est plus susceptible d’être utilisé par les utilisateurs (jeux, applications gratuites, etc.). Après avoir pénétré dans le système, le virus reste inactif jusqu’à ce que le programme infecté soit lancé et quand il commence, il commence à affecter d’autres programmes du système. Les virus sont de types différents en fonction de leurs propriétés et des procédés d’activation. Prévenir un virus de pénétrer dans un système sur la première place est une solution idéale pour protéger un système à partir d’une attaque de virus qui est très difficile, voire impossible, de le faire à l’époque des systèmes en réseau. Donc, la meilleure alternative à la protection de virus pourrait être la détection d’un virus, une fois détecté identifier son type, et ensuite essayer de le retirer du système ou en cas d’impossibilité retirer l’ensemble du programme concerné. Celles-ci sont réalisées par des programmes spéciaux connus sous le nom antivirus. Les programmes antivirus sont utilisés pour protéger le système d’être affectés par des virus. Un antivirus maintient définition de virus qui est le code pur des virus. L’antivirus inspecte ensuite les fichiers exécutables du système pour vérifier s’il y a un code correspondant à la définition de virus. Si oui, ils essaient de fixer le programme en supprimant le code du virus.

Un worm est similaire à un virus, sauf qu’il a la capacité de se répliquer et n’a pas besoin de se fixer sur d’autres programmes à se reproduire. C’est un programme indépendant qui se propage via des connexions réseau.

**Conclusion**

Les attaques du système peuvent être soit internes (arrive après que l’utilisateur se connecte avec succès) ou externe à partir d’un endroit éloigné en particulier dans les systèmes en réseau. Le cheval de Troie, la bombe logique, l’usurpation d’identité, dépassement de mémoire tampon sont quelques-unes des attaques de l’intérieur du système alors que les virus, les vers, les codes mobiles sont des attaques provenant de l’extérieur du système.

**Évaluation**

Comment le débordement de tampon attaque-t-il un système?

Donner les solutions pour protéger le système contre les attaques de vers?

Quelle est la différence entre un virus et un ver ?

Qu’est-ce qu’un cheval de Troie et comment attaque-t-il le système ?

Quelle est la différence entre une attaque interne et une attaque externe ?

**Les mécanismes de protection**

Introduction

Un système d’exploitation a la responsabilité de faire respecter les mécanismes de protection de sécurité pour éviter les menaces possibles de se produire et de protéger les différents afin de protéger les différentes ressources du système. Chaque actif d’un système a un nom et un ensemble bien défini d’opérations à travers laquelle l’accès est fait. Dans cette section, nous abordons quelques contre-mesures fixées par un système informatique pour se protéger contre les attaques de sécurité et les principes de conception qui doivent être suivies pour renforcer la sécurité de système. Tel qu’un système d’exploitation est conçu, chaque objet (utilisateur, processus) a un ensemble de privilèges qui doivent être garantie. Les mécanismes de sécurités doivent être interne et garantir l’accès contrôlé pendant toutes les opérations. Le système de sécurité doit être revu de manière continuelle.Ci-dessous, nous explorons les principaux mécanismes de sécurité.

**La cryptographie**

Un message est significatif quand il est lisible (écrit dans des caractères que l’on comprend). La cryptographie est un mécanisme qui consiste à transformé une donnée (un texte par exemple) qui était alors claire (facile à déchiffrer) en une donnée (code) difficile à déchiffrer. Cela suppose qu’à la réception du message, le destinataire doit être à mesure de décrypter le message. La transformation des données à transmettre dans des codes indéchiffrables (pour les non destinataires) est appelé le cryptage tandis que la transformation du code reçu en le message d’origine s’appelle le décryptage.

**Authentification des utilisateurs**

Un autre mécanisme de sécurité est l’authentification. Pour le système chaque utilisateur possède une identification unique. L’authentification des utilisateurs est un processus de détermination de l’identité d’un utilisateur sur la base duquel son / ses activités autorisées sont également identifiés. Il permet à une entité (un utilisateur ou un système) de prouver son identité à une autre entité. Les éléments d’identifications d’un utilisateur peuvent être : Nom utilisateur et mot e passe, données biométriques (empreintes digitales, l’iris, …..), etc.

Pour les données non biométriques en l’occurrence le nom utilisateur et le mot de passe, il est conseillé de les changer régulièrement. Il est conseillé de créer des mots de passe long, comprenant plusieurs types de caractères (lettes, chiffres, caractères spéciaux, majuscules minuscules). Remarque : Pour un mot de passe compliqué, les utilisateurs ont tendance à les écrire soit dans un carnet sur la table, les coller sur les murs etc. ; D’où ce qui était une force devient une faiblesse. Les utilisateurs doivent être sensibilisés pour ce qui est de la sécurité du système informatique de leur organisation.

Enfin notons également que le contrôle d’accès physique est un élément important des mécanismes de sécurité du système informatique.

**Conclusion**

Un système informatique a besoin de se protéger contre les accès non autorisés et illégitime. Pour ce faire, il définit divers mécanismes de protection du système dont la cryptographie, l’authentification des utilisateurs et sans doute aussi le contrôle d’accès physique aux équipements.

**Évaluation**

En quoi consiste la cryptographie?

Quelles sont les règles à tenir en compte dans le choix d’un mot de passe. Donner l’inconvénient principal d’un mot de passe complexe ?