

TP1 – Systèmes d’exploitation (Windows, MacOs, Linux)

Par  
Frédérik Boutin

Travail présenté à Mme Lise Boudreault

Dans le cadre du cours Systèmes d’exploitationINF34207-MS

14 février 2024

Table des matières

[1. Introduction 2](#_Toc158833385)

[1.1 Dans un tableau, présentez les caractéristiques (lignes) de l’ordinateur que vous utilisez pour ce cours : 2](#_Toc158833386)

[1.2 Lorsque vous utilisez votre ordinateur, décrivez au moins deux applications (p.ex. un jeu vidéo, un outil de montage, un outil de modélisation 3D, etc.) : 3](#_Toc158833387)

[2. Programmation 4](#_Toc158833388)

[2.1 Choisissez trois langages de programmation (ex: C, C++, C#, JavaScript, Java, Python) : 4](#_Toc158833389)

[2.2 Présentez le code requis pour effectuer trois requêtes vers le système d’exploitation : 4](#_Toc158833390)

[Saisie d’une information (entrée de données au clavier) 4](#_Toc158833391)

[Affichage de l’information à l’écran 5](#_Toc158833392)

[Impression de la même information qu’à l’étape précédente, mais à partir d’une imprimante, donc, effectuer un appel à l’imprimante (par défaut) 7](#_Toc158833393)

[2.3 Au niveau de la complexité de la programmation, que remarquez-vous entre ces trois langages ? 10](#_Toc158833394)

[3. Bases de données 10](#_Toc158833395)

[3.1 Quel est le rôle du système d’exploitation (séquence d’étapes) lorsqu’un langage de programmation transmet une requête vers un serveur de base de données (ex : MySQL)? 10](#_Toc158833396)

[Médiagraphie 12](#_Toc158833397)

[Annexe 13](#_Toc158833398)

# Introduction

## 1.1 Dans un tableau, présentez les caractéristiques (lignes) de l’ordinateur que vous utilisez pour ce cours :

Avant-propos : Il est à noter que pour ce travail, j’ai utilisé mon ordinateur portable. Lors de la séance Zoom, j’avais inscrit dans le chat les caractéristiques de mon pc de type tour de bureau. Ainsi, il est normal qu’il ne s’agisse pas des même caractéristiques.

|  |  |
| --- | --- |
| **Caractéristique de l’ordinateur** | **Valeur de la caractéristique** |
| Nom du processeur | 11th Gen Intel(R) Core(TM) i7-1165G7 @ 2.80GHz, 2803 Mhz, 4 Core(s), 8 Logical Processor(s) |
| Manufacturier du processeur | Intel Corporation |
| Vitesse de base du processeur | 2.80GHz |
| Nombre de cœur du processeur | 4 |
| Taille de la cache de niveau 1 | 320 KB |
| Taille de la cache de niveau 2 | 5.0 MB |
| Taille de la cache de niveau 3 | 12 MB |
| Capacité de la mémoire vive | 16.0 GB |
| Vitesse de la mémoire vive | 3200 MHz |
| Type de mémoire vive | DDR4 |
| Nom du disque dur principal (modèle) | OS (C :) |
| Manufacturier du disque dur principal | (Standard disk drives) |
| Capacité du disque dur principal | 475 GB |
| Système de fichiers utilisé du disque dur principal | NTFS (New Technology File System) |
| Type de connexion du disque dur principal (SATA, USB, NVMe, etc.) | NVMe INTEL SSDPEKNW512G8 |
| Nom de la carte graphique (si présente) | Mon ordinateur n’a pas de carte graphique, cependant il a une solution graphique intégrée, soit Intel Iris Xe Graphics. Cela signifie que l’unité de traitement graphique (GPU) est intégrée directement dans la puce du processeur, partageant des ressources telles que la mémoire avec le CPU. |
| Manufacturier de la carte graphique | Intel Corporation (manufacturier de la solution graphique intégrée ≠ carte graphique) |
| VRAM de la carte graphique | Intel Iris Xe Graphics dispose de VRAM, mais il ne s’agit pas d’une mémoire physique séparée comme pour les cartes graphiques. Au lieu de cela, elle utilise une partie de la mémoire principale (RAM) du système comme mémoire vidéo partagée. |
| Vitesse de l’horloge de la carte graphique | Après avoir essayé sans succès de trouver ce renseignement utilisant Task Manager et Intel Graphics Command Center, je n’ai pas trouvé. Il faudrait utiliser un logiciel tiers, mais je ne préfère pas m’y risquer. |

Les outils que j’ai utilisés pour trouver les différentes informations sont les suivant :

* System Information
* Task Manager
* Device Manager
* Windows Power Shell en mode admin (commande Get-CimInstance -ClassName Win32\_PhysicalMemory | Format-Table SMBIOSMemoryType pour obtenir le type de mémoire vive)

## 1.2 Lorsque vous utilisez votre ordinateur, décrivez au moins deux applications (p.ex. un jeu vidéo, un outil de montage, un outil de modélisation 3D, etc.) :

La première application est Google Chrome (version : 120.0.6099.225) qui est mon navigateur web. Cette application est plus efficiente en termes de temps de réponse, puisque les navigateurs modernes sont conçus pour optimiser leur temps de réponse en fonction des ressources matérielles disponibles, telles que la mémoire RAM et les processeurs multicœurs. Parmi les exemples de moyens d’optimisation, il y a l’architecture multiprocessus qui permet une meilleure isolation entre les différents processus de l’application et une meilleure répartition des tâches sur plusieurs processus (temps de réponse plus rapide), l’utilisation d’une programmation asynchrone pour la gestion des tâches, une gestion intelligente des ressources qui permet l’attribution de priorité aux onglets actifs et l’utilisation du chargement différer pour les ressources non prioritaire. Il faut noter également l’utilisation d’un moteur de rendu optimisé (patron de conception Composite) afin de permettre la gestion des pages par arborescences d’objet, ce qui améliore le temps d’opération sur les éléments des pages. Durant un test d’utilisation pour la lecture d’un vidéo YouTube (musique), l’application utilisait 2-5% du CPU, 650-750 MB de RAM, 0-0.1% du disque et la lecture était fluide.

La seconde application est mon logiciel gratuit d’antivirus AVG Internet Security System (version 23.12.8700.0). Cette application, lorsqu’elle effectue une analyse du système, doit accéder à plusieurs fichiers du système et effectuer des opérations de comparaison pour détecter des menaces. Ainsi, tout ce processus occasionne une grande utilisation des ressources matérielles de l’ordinateur, autant au niveau des calculs du processeur qu’au niveau du disque pour l’accès aux fichiers. On sait que l’accès au disque est la forme d’opération la plus lente sur une machine et l’antivirus doit y accéder souvent pour son analyse, ainsi il est possible de comprendre pourquoi elle possède un temps d’exécution plus lent. Un test d’analyse intelligente a occasionné des utilisations de 20-70% du CPU, 150-300 MB de RAM et 1.5-21% du disque.

En résumé, comme il a été possible de le voir à travers ces deux applications, la différence entre l’efficience d’exécution d’une application va beaucoup dépendre des moyens et des principes de conception qu’elle utilise pour utiliser au mieux les ressources matériels de la machine, le type de ressource qu’elle doit utiliser sur la machine, ainsi que la possibilité ou non d’optimiser les processus. Ainsi, un navigateur web qui effectue une bonne séparation de tâches peut les exécuter plus rapidement et en parallèle, son utilisation d’une cache et du du chargement différer permettront un accès plus rapide aux informations de page. À l’opposé, une application antivirus doit nécessairement faire son analyse de fichiers en passant par le régulièrement par l’accès au disque et monopolise beaucoup le processeur, ce qui augmente le temps de traitement.

# Programmation

## 2.1 Choisissez trois langages de programmation (ex: C, C++, C#, JavaScript, Java, Python) :

J’ai choisi les langages Python, C# et C++. Les images ci-dessous présentent les trois codes de ces langages. Les trois programmes suivent la même logique, c’est-à-dire qu’ils font des appels de fonctions que j’ai créé pour représenter la saisie d’information, l’affichage à l’écran et l’impression à l’imprimante. Pour chacun des cas, la fonction de saisie ce nomme EnteringInformation, la fonction d’affichage à l’écran ce nomme DisplayInformation et finalement, la fonction se nomme Printing.

## 2.2 Présentez le code requis pour effectuer trois requêtes vers le système d’exploitation :

Avant-propos : Voici le code des fonctions permettant d’effectuer les requêtes demandées. Il est à noter que les codes des appels à ces fonctions (Main) ainsi que les images des résultats console ont été placé en annexe.

### Saisie d’une information (entrée de données au clavier)



Figure 1 - Code Python pour une saisie



Figure 2 -Code C# pour une saisie



Figure 3 -Code C++ pour la saisie

### Affichage de l’information à l’écran



Figure 4 -Code Python pour l'affichage



Figure 5 -Code C# pour l'affichage



Figure 6 -Code C++ pour l'affichage

### Impression de la même information qu’à l’étape précédente, mais à partir d’une imprimante, donc, effectuer un appel à l’imprimante (par défaut)



Figure 7 -Code python pour l'impression



Figure 8 -Code C# pour l'impression



Figure 9 -Code C++ pour l'impression

## 2.3 Au niveau de la complexité de la programmation, que remarquez-vous entre ces trois langages ?

Avant-propos : Il est à noter que le code C#, pour l’impression à l’imprimante, utilise une bibliothèque graphique faisant partie du système d’exploitation de Windows, contrairement aux codes Python et C++ qui utilisent les mêmes fonctions liées à l'impression sous Windows.

Il est possible de remarquer que plus le langage de programmation est de bas niveau, autrement dit, plus le langage est près des instructions machines, plus la programmation est complexe. Si l’on compare les deux codes Python (code haut niveau) et C++ (code bas niveau) pour l’impression avec l’imprimante par exemple, il est possible de voir que le code Python ne nécessite que 10 lignes de code, alors que celui en C++ nécessite 39 lignes. De plus, l’implémentation des fonctions sous Python est beaucoup plus simple, du fait que les paramètres qui les composent sont plus directement accessible. Il est donc possible d’en conclure que bien que les deux codes utilisent les mêmes fonctions d’imprimerie pour interagir avec le système d’impression de Windows, il y a un certain nombre d’opérations intermédiaires effectuées en arrière-plan par l’interpréteur du code Python pour arriver au même résultat que le code C++. Après tout, puisque le système d’exploitation est l’interface permettant aux applications de communiquer avec le pilote de l’imprimante (par la suite, du pilote à l’imprimante), il doit donc recevoir les mêmes instructions d’entrée si on veut obtenir un même résultat. Ce sont précisément les opérations intermédiaires d’arrière-plan qui rend le langage Python plus facile à utiliser, cependant cela peut représenter un inconvénient lorsqu’on veut obtenir un plus grand contrôle de programmation puisqu’on ne sait pas toujours tout ce qui est fait en arrière-plan.

# Bases de données

## 3.1 Quel est le rôle du système d’exploitation (séquence d’étapes) lorsqu’un langage de programmation transmet une requête vers un serveur de base de données (ex : MySQL)?

Le rôle du système d’exploitation est celui d’un intermédiaire entre le langage de programmation (application du langage) et le serveur de base de données. Il va ainsi s’occuper des opérations de création et de destruction du réseau de communication entre les deux entités, de gestion de ce dernier, ainsi que de la transmission des données. Voici un résumé de la séquence d’étapes :

1. Le code de l’application (côté client) utilise l’interface de programmation d’application ou API (via une librairie par exemple) propre à la base de données, permettant ainsi l’interaction avec son système de gestion (SGBD), pour transmettre une requête.
2. Il y a transformation de l’appel API de haut-niveau en plusieurs appels de plus bas-niveau que le système d’exploitation peut comprendre. Ces appels bas-niveau comprennent les directives pour l’établissement des communications entre les applications.
3. Lorsque le système d’exploitation aura donné la priorité d’exécution à l’application côté client, il va exécuter ces instructions bas niveau tout en allouant les ressources nécessaires pour effectuer la communication entre l’application cliente et l’application côté serveur.
4. Le système d’exploitation va créer un réseau de communication entre l’application cliente et le serveur de base de données, il va donc ouvrir les ports de communication.
5. La requête initiale est ainsi adaptée au formatage de la base de données. Le système d’exploitation la transmet par le réseau de communication précédemment crée.
6. Le serveur de base de données va ensuite traiter la requête, suit à sa réception par son interface réseau. Une fois la requête traitée, le serveur va renvoyer la réponse sur le port du réseau de communication crée initialement par le système d’exploitation de l’application client.
7. Le système d’exploitation gère la transmission des données de la réponse de retour via le réseau de communication jusqu’à l’application cliente en cours.
8. À la fin de l’échange, le système d’exploitation ferme les ports de communication et relâche les ressources y étant associées.

# Médiagraphie

*Chargement différé des images au niveau du navigateur pour le Web | Articles*. (s. d.). web.dev. Consulté 14 février 2024, à l’adresse https://web.dev/articles/browser-level-image-lazy-loading?hl=fr

*Chromium Docs—Threading and Tasks in Chrome*. (s. d.). Consulté 14 février 2024, à l’adresse https://chromium.googlesource.com/chromium/src/+/main/docs/threading\_and\_tasks.md

Client Server Communication in Operating System. (2023, décembre 26). *GeeksforGeeks*. https://www.geeksforgeeks.org/client-server-communication-in-operating-system/

Collins, E. (2021, février 23). *Does Antivirus Software Slow Down Your Computer?* MUO. https://www.makeuseof.com/does-antivirus-slow-down-computer/

dotnet-bot. (s. d.). *Graphics Class (System.Drawing)*. Consulté 14 février 2024, à l’adresse https://learn.microsoft.com/en-us/dotnet/api/system.drawing.graphics?view=dotnet-plat-ext-8.0

drewbatgit. (2021, janvier 7). *Print Spooler API - Win32 apps*. https://learn.microsoft.com/en-us/windows/win32/printdocs/print-spooler-api

Le, J. (2020, juin 9). *How Operating Systems Work : 10 Concepts you Should Know as a Developer*. Medium. https://data-notes.co/how-operating-systems-work-10-concepts-you-should-know-as-a-developer-8d63bb38331f

Why Does Google Chrome Have Multiple Processes Running. (2024, février 5). *Robots.Net*. https://robots.net/software-and-applications/browsers-and-extensions/why-does-google-chrome-have-multiple-processes-running/

# Annexe



Figure 10 - Code Python d'appel de fonctions (Main)

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Figure 11 - Capture d'écran du terminal du code Python



Figure 12 - Code C# d'appel de fonctions (Main)

Une image contenant texte, capture d’écran, noir

Description générée automatiquement

Figure 13 - Capture d'écran de la console du code C#



Figure 14 - Code C++ d'appel de fonctions (Main)

Une image contenant texte, capture d’écran, Police

Description générée automatiquement

Figure 15 - Capture d'écran de la console du code C++