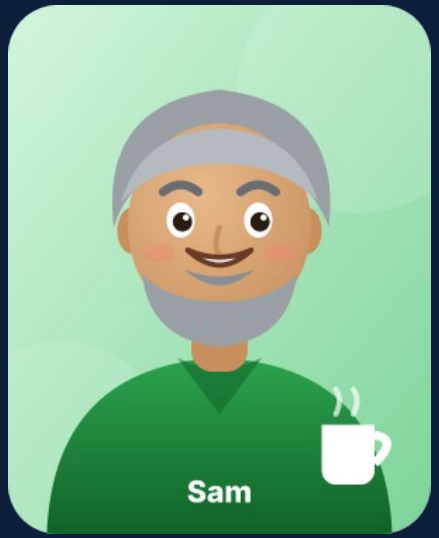
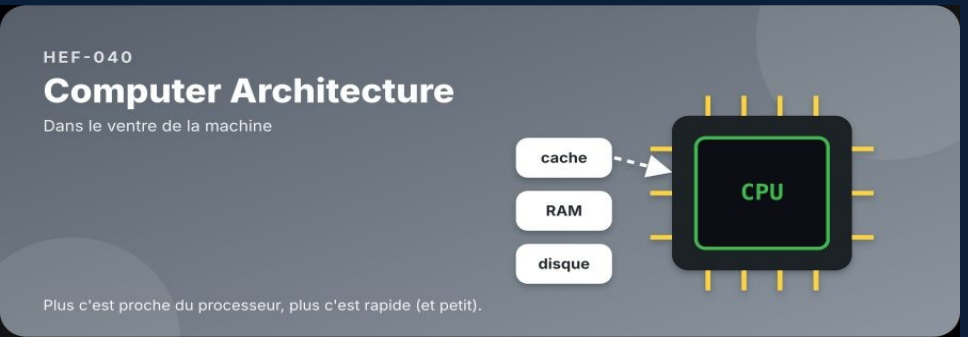
HEF-040 · MISSION 07 / 10

Bus, interruptions et périphériques

Un ordinateur est une équipe : processeur, mémoire, disques, cartes… Comment communiquent-ils ?





Badge : Chef de gare

70 min · débutant

L'histoire

Kodi branche une clé USB. Une fraction de seconde plus tard, une fenêtre s'ouvre. « Comment l'ordinateur sait-il qu'il y a une clé ? Il passe son temps à vérifier ? »



> *« Non, dit Sam. Le périphérique frappe à la porte du processeur : c'est une interruption. »*

Tu découvres comment les composants d'un ordinateur dialoguent sans que le processeur perde son temps.

Ta mission

À la fin, je serai capable de…

1. décrire le rôle d'un bus (données, adresses, contrôle)
2. comparer l'attente active (polling) et les interruptions
3. expliquer l'accès direct à la mémoire (DMA)
4. citer des exemples de bus et de périphériques
5. calculer la bande passante d'un bus

Défi d'échauffement

Défi d'échauffement

Tu attends un colis. Option A : tu vas vérifier la boîte aux lettres toutes les 5 minutes. Option B : le livreur sonne à ta porte. Quelle option te fait perdre le moins de temps ? Dans quel cas l'option A reste-t-elle raisonnable ?

Le concept

Les bus

Un **bus** est un ensemble de fils (ou de liaisons) qui transporte de l'information entre composants. On distingue trois types de lignes :

Ligne	Rôle
Adresses	qui est concerné (quelle case mémoire, quel périphérique)
Données	l'information échangée
Contrôle	lire ou écrire, horloge, signaux

Exemples : le bus mémoire (CPU ↔ RAM), **PCIe** (cartes graphiques, SSD NVMe), **USB** (périphériques externes), **SATA** (disques).

Le concept

Bande passante

La **bande passante** est la quantité de données transférée par seconde :

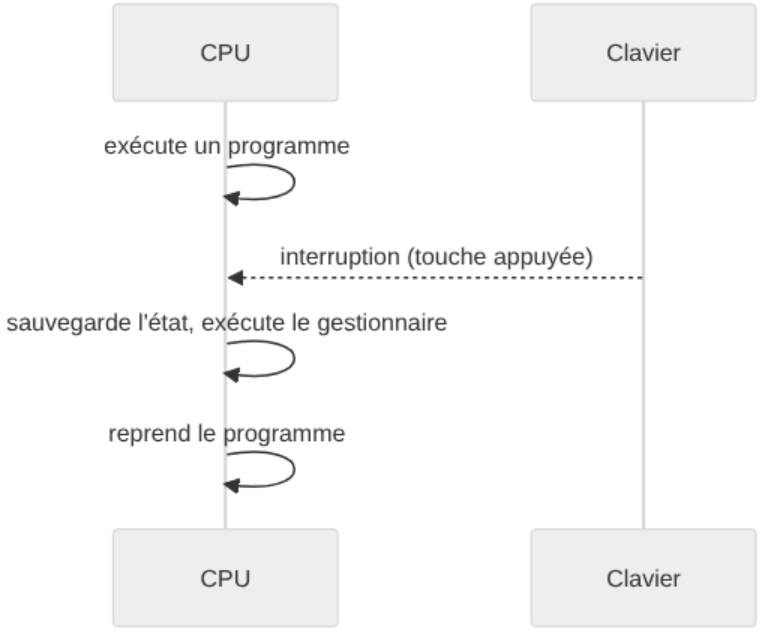
bande passante = largeur du bus (en octets) × fréquence de transfert

Un bus de 8 octets de large à 1 milliard de transferts par seconde offre 8 Go/s.

Polling ou interruption

- **Polling** (attente active) : le processeur **demande régulièrement** « as-tu fini ? ». Simple, mais il perd du temps.
- **Interruption** : le périphérique **envoie un signal** quand il a quelque chose à signaler. Le CPU interrompt son travail, exécute un petit programme (*gestionnaire d'interruption*), puis reprend.

Polling ou interruption (suite)



Le concept

Polling ou interruption (suite)

Description du schéma : le CPU exécute un programme. Le clavier envoie une interruption. Le CPU sauvegarde son état, exécute le gestionnaire correspondant, puis reprend le programme où il en était.

Le DMA

Sans aide, le CPU devrait copier lui-même chaque octet venant d'un disque vers la RAM. Avec le **DMA** (*Direct Memory Access*), un contrôleur spécialisé fait la copie pendant que le CPU travaille ailleurs. Le CPU est averti par une interruption à la fin.

Pilotes

Un **pilote** (*driver*) est le logiciel qui sait parler à un périphérique précis. Le système d'exploitation s'appuie dessus : tu écris dans un fichier, le pilote du disque s'occupe du reste.

Le concept

Pilotes (suite)

🧠 ***Astuce*** *: interruptions et DMA permettent de faire* ***plusieurs choses en même temps*** *(le CPU calcule pendant que le disque lit).*

Exemple pas à pas

Problème : comparer le coût CPU du polling et des interruptions pour attendre un périphérique qui répond après 10 000 cycles.

```
def polling(delai_reponse, cout_verification=20):
    cycles = 0
    while cycles < delai_reponse:
        cycles += cout_verification        # le CPU teste le périphérique en boucl
   e
    return cycles, cycles // cout_verification

def interruption(delai_reponse, cout_gestionnaire=500):
    # le CPU fait autre chose pendant l'attente, puis paie le coût du gestionnaire
    cycles_libres = delai_reponse
    return cycles_libres, cout_gestionnaire

c_poll, tests = polling(10_000)
libres, cout = interruption(10_000)
print("polling      : CPU occupé", c_poll, "cycles pour", tests, "vérifications, 0
```

Exemple pas à pas (suite)

```
    cycle libre")
print("interruption : CPU libre", libres, "cycles, coût du gestionnaire", cout, "c
   ycles")
```

Lecture : avec le polling, le CPU consomme tout son temps à vérifier. Avec une interruption, il paie seulement un coût fixe (le gestionnaire) et utilise les 10 000 cycles à autre chose.
Mais pour un périphérique qui répond **presque tout de suite**, le polling est moins cher que le coût d'une interruption : le bon choix dépend du délai de réponse.

Calcul de bande passante :

```
largeur_octets = 8
transferts_par_seconde = 3_200_000_000        # 3,2 GT/s
print(largeur_octets * transferts_par_seconde / 1e9, "Go/s")   # 25.6
```

À toi, avec aide

Un processeur à 3 GHz (3 milliards de cycles par seconde) reçoit des octets d'un périphérique à 1 Mo/s.

1. Combien de cycles le CPU a-t-il entre deux octets ?
2. Si le gestionnaire d'interruption coûte 2 000 cycles, quelle fraction du temps du CPU consomme-t-il si une interruption survient **à chaque octet** ?
3. Que se passe-t-il si le périphérique passe à 100 Mo/s ? Quelle solution adopter (DMA, regroupement) ?

À toi, avec aide · réponse

Indices progressifs

Indice 1

1 Mo/s ≈ 1 000 000 octets par seconde.

Indice 2

Cycles entre deux octets = $3 \times 10^9 / 10^6$.

Indice 3

À 100 Mo/s, on obtient moins de 30 cycles entre deux octets.

À toi, avec aide · réponse

Réponse

```
cycles_par_s = 3_000_000_000
for debit in (1_000_000, 100_000_000):
    entre = cycles_par_s / debit
    fraction = 2000 / entre
    print(debit, "octets/s :", entre, "cycles entre octets ; gestionnaire =", f"{1
   00 * fraction:.0f} % du CPU")
# 1 000 000 -> 3000 cycles ; 67 %
# 100 000 000 -> 30 cycles ; 6667 % (impossible)
```

À 1 Mo/s, une interruption par octet monopolise déjà 2/3 du processeur. À 100 Mo/s, c'est impossible. La solution : le **DMA** transfère des blocs entiers et ne déclenche qu'**une** interruption à la fin du bloc.

À toi, en solo

Écris un petit **simulateur d'événements** avec une file :

- un périphérique « clavier » génère des événements (touche, instant) ;
- le CPU exécute une tâche de fond (compter) ;
- à chaque événement, le CPU interrompt sa tâche, traite l'événement (coût 100 unités) puis reprend.

Compare avec une version polling qui teste la file toutes les 10 unités de temps. Mesure : temps de réaction moyen aux événements et travail de fond accompli sur 10 000 unités.

À toi, en solo · réponse

Pistes de correction

```
import random

random.seed(2)
evenements = sorted(random.sample(range(10000), 20))    # instants des touches
COUT_TRAITEMENT = 100

def avec_interruptions():
    t = travail = 0
    reactions = []
    suivants = list(evenements)
    while t < 10000:
        if suivants and suivants[0] <= t:
            reactions.append(t - suivants.pop(0))
            t += COUT_TRAITEMENT
        else:
            t += 1
            travail += 1
    return travail, sum(reactions) / len(reactions)
```

À toi, en solo · réponse

Pistes de correction (suite)

```
def avec_polling(periode=10, cout_test=1):
    t = travail = 0
    reactions = []
    suivants = list(evenements)
    while t < 10000:
        if t % periode == 0:
            t += cout_test
            while suivants and suivants[0] <= t:
                reactions.append(t - suivants.pop(0))
                t += COUT_TRAITEMENT
        else:
            t += 1
            travail += 1
    return travail, (sum(reactions) / len(reactions) if reactions else None)

print("interruptions :", avec_interruptions())
print("polling       :", avec_polling())
```

Pistes de correction (suite)

L'interruption réagit tout de suite (temps de réaction ≈ 0) ; le polling réagit en moyenne après la moitié de la période et consomme du temps de test en plus.

Pièges classiques

Piège	Pourquoi c'est un problème	Comment l'éviter
Confondre bus et câble	Un bus est un protocole + des lignes partagées	Penser « canal de communication »
Utiliser le polling pour un événement rare	CPU gaspillé	Préférer les interruptions
Ignorer le coût d'une interruption	Trop d'interruptions saturent le CPU	Regrouper (DMA, traitement par blocs)
Croire qu'un périphérique est instantané	Latence et débit existent	Mesurer ou lire les spécifications

Mini-boss

Chaîne d'un fichier. Décris et chiffre ce qui se passe quand ton programme lit un fichier de 100 Mo sur un SSD NVMe :

1. Les composants impliqués (CPU, RAM, contrôleur, SSD, bus PCIe) et les bus utilisés.
2. Qui copie les données : le CPU ou le DMA ? Pourquoi ?
3. Combien d'interruptions si le transfert se fait par blocs de 1 Mo ? par blocs de 4 Ko ?
4. Le débit du bus PCIe 3.0 x4 est d'environ 3,9 Go/s. Combien de temps minimum pour 100 Mo ?
5. Le débit du disque est-il le facteur limitant ? Quels autres facteurs ?

Mini-boss · réponse

Pistes de correction

```
taille = 100 * 1024 * 1024
for bloc in (1024 * 1024, 4096):
    print(bloc, "octets/bloc →", taille // bloc, "interruptions")
debit = 3.9e9
print("temps minimum :", round(taille / debit * 1000, 1), "ms")
```

Avec des blocs de 1 Mo : 100 interruptions ; avec 4 Ko : 25 600. Le DMA copie les données vers la RAM sans le CPU. Temps minimum sur le bus : environ 27 ms. Les facteurs limitants possibles : vitesse réelle du SSD, système de fichiers, cache de pages, logiciel qui lit les données.

Quiz éclair

Quiz éclair

1. Une interruption est :

A. un signal d'un périphérique vers le CPU
B. une panne
C. un type de mémoire

2. Le DMA permet :

A. de chiffrer des fichiers
B. de copier des données sans mobiliser le CPU
C. d'augmenter la fréquence du CPU

3. Un pilote (driver) est :

A. un composant matériel

Quiz éclair

Quiz éclair (suite)

B. un type de bus
C. le logiciel qui sait parler à un périphérique

4. 🔁 *Révision mission 06* : Écrire 1 octet un million de fois est généralement :

A. bien plus lent qu'écrire 1 Mo d'un coup
B. aussi rapide
C. impossible

5. 🔁 *Révision mission 05* : La localité spatiale signifie :

A. une donnée récente sera réutilisée
B. les données voisines sont souvent utilisées ensemble
C. la mémoire est proche du CPU

Quiz éclair · réponse

Réponses

1. **A**.
2. **B**.
3. **C**.
4. **A**.
5. **B**.

Je réfléchis

Je réfléchis

1. Quels objets de ta vie fonctionnent par « interruption » (alarme, notification) et lesquels par « polling » ?
2. Pourquoi les notifications excessives peuvent-elles être comparées à trop d'interruptions ?

À retenir

À retenir

1. Un bus transporte adresses, données et contrôle.
2. Bande passante = largeur × fréquence de transfert.
3. Polling : le CPU demande ; interruption : le périphérique appelle.
4. DMA : copie sans le CPU.
5. Pilote = traducteur entre OS et matériel.

MISSION ACCOMPLIE

+100 XP

Badge débloqué : Chef de gare

Prochaine mission : Un programme en cours d'exécution

