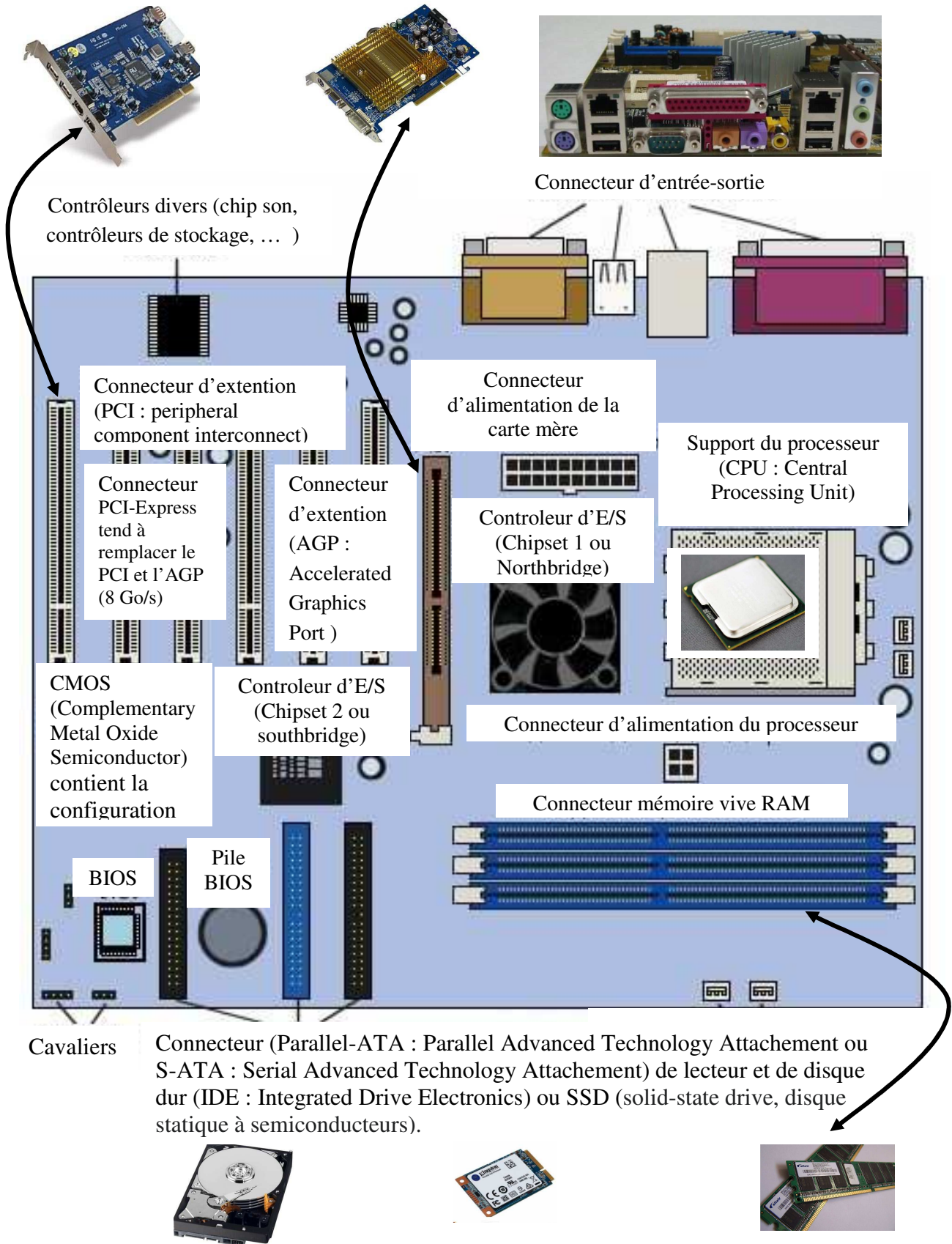
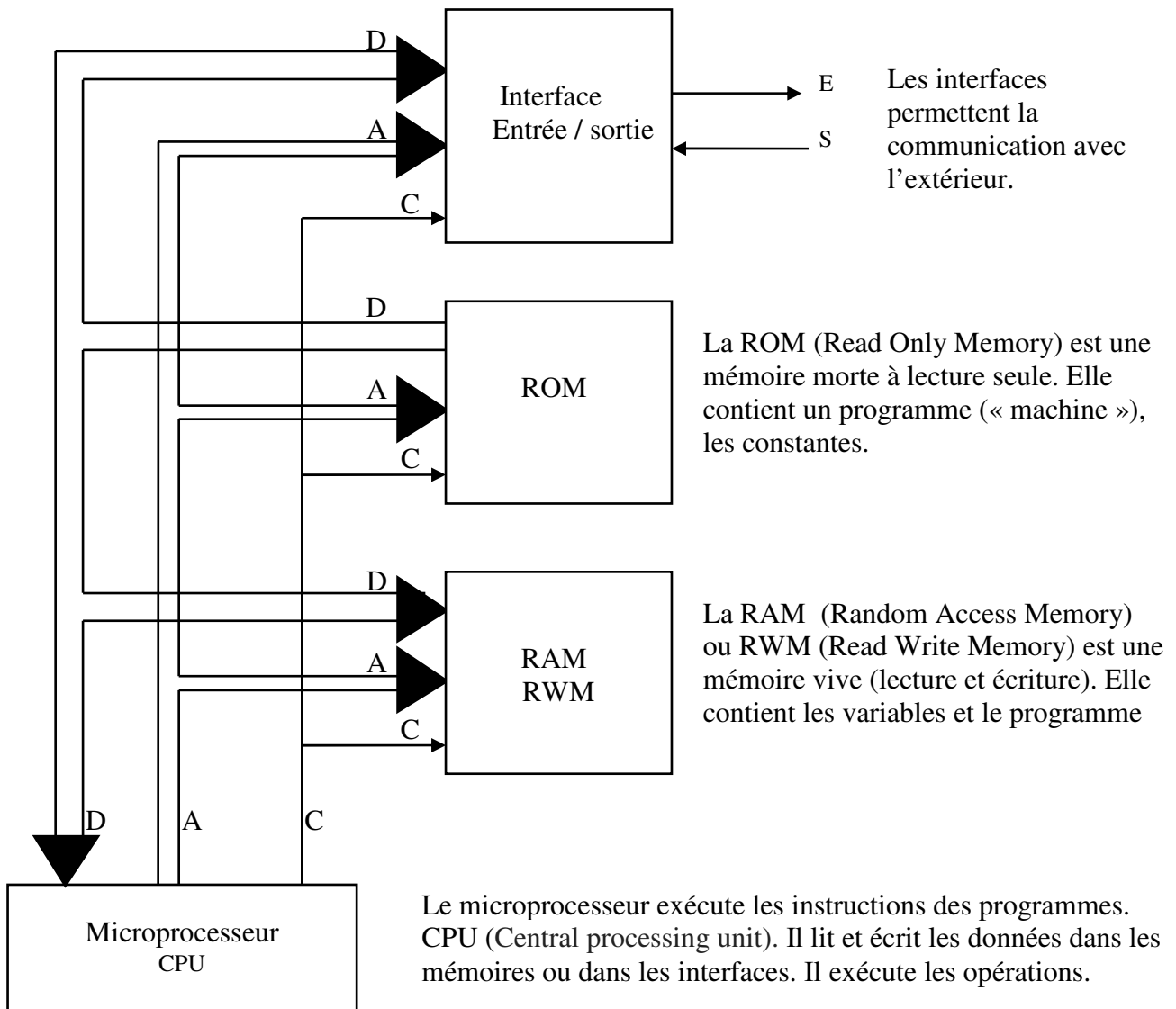


ARCHITECTURES MATERIELLES, SYSTEMES D'EXPLOITATION ET RESEAUX

1. ÉLÉMENTS D'ARCHITECTURE (rappels de classe de première)

1.1. LA CARTE MERE (Motherboard)



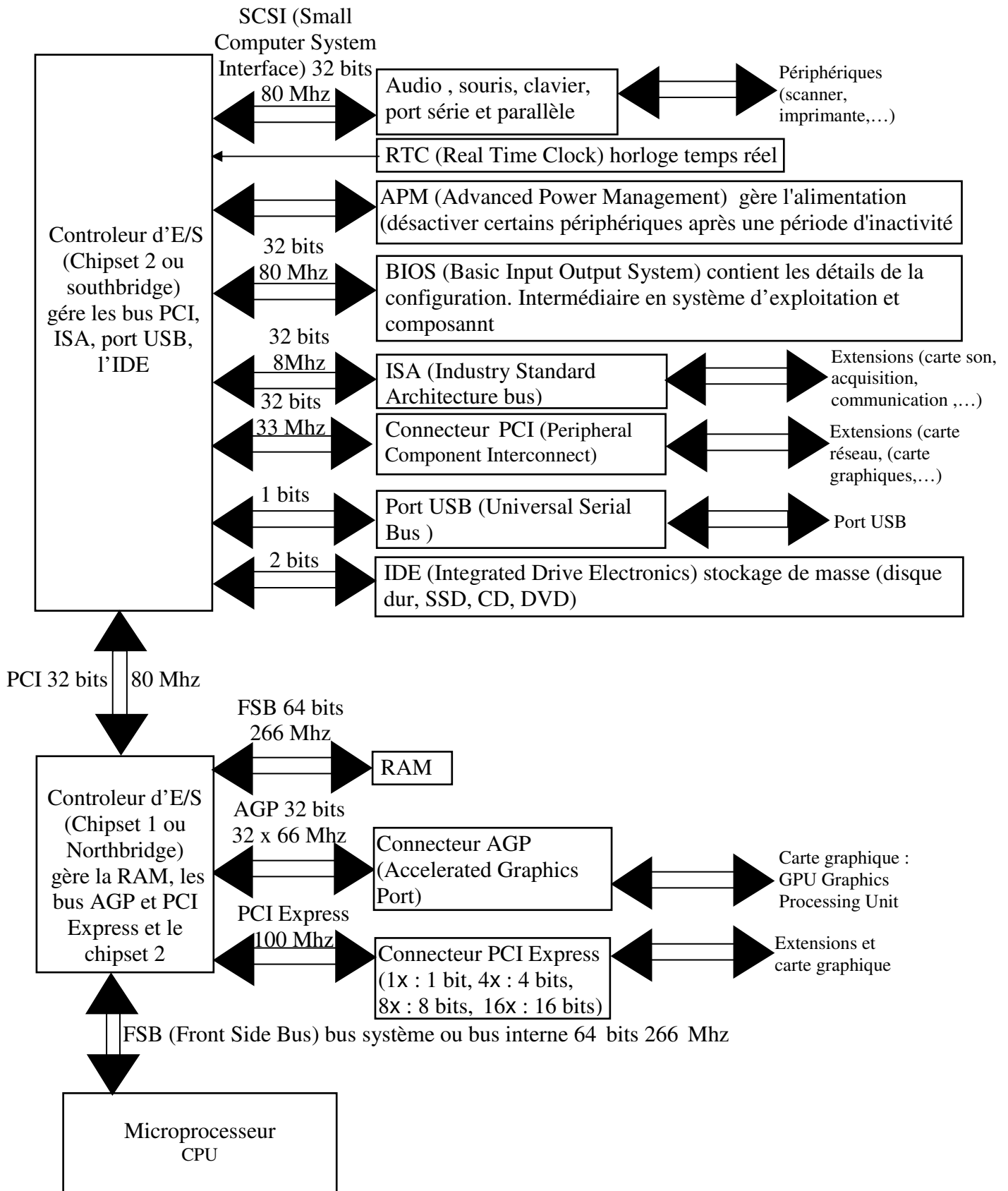
1.2. PRINCIPE DE FONCTIONNEMENT DE L'UNITE CENTRALE

Les différents constituants (CPU, RAM, ROM, Interfaces) sont reliés par des BUS :

A = Adresses (unidirectionnel)

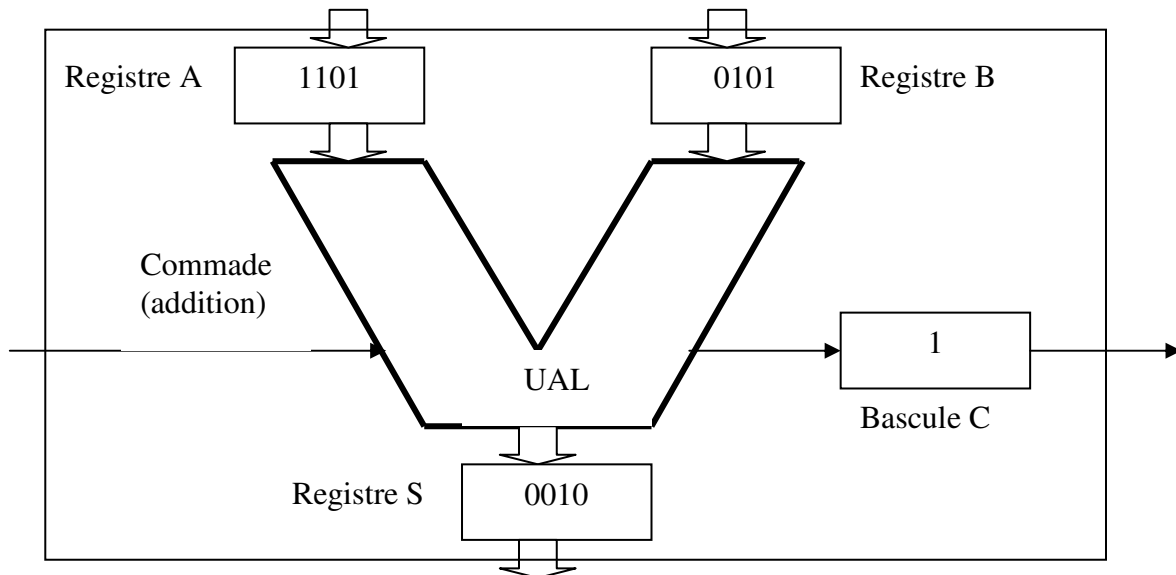
D = Données (bidirectionnel)

C = Contrôle (unidirectionnel)

1.3. CONTROLEURS D'ENTREE/SORTIE

1.4. LE MICROPROCESSEUR PRINCIPE

Entre autre, le microprocesseur contient une unité arithmétique et logique ALU ou UAL Cet organe permet la réalisation d'opérations arithmétiques et logiques. L'ALU possède un additionneur n bits et des fonctions logiques. Dans l'exemple ci-dessous, une addition est effectuée entre le registre A et le registre B. Le résultat est mis dans le registre S et la retenue dans la bascule C.



1.5. LES MEMOIRES PRINCIPE

Données	D7	D6	D5	D4	D3	D2	D1	D0
Adresses								
\$ 1004								
\$1003								
\$1002								
\$1001								
\$1000								

Les mémoires sont des « cases » contenant des données binaires.

L'emplacement de chaque « case » est défini par son adresse.

Le microprocesseur peut :
lire les mémoires mortes
lire et écrire les mémoires vives

1.5.1. DIFFERENTS TYPES DE MEMOIRES

1.5.1.1. MEMOIRES DE MASSE

Disque dur, disquette, bande magnétique, SSD, CDROM, DVD, ZIP

1.5.1.2. MEMOIRES A SEMI - CONDUCTEUR

Mémoires à lecture seule, stockent les données sous forme de ddp (différence de potentiel) ou de courant même après coupure de l'alimentation :

- ROM : Read Only Memory, mémoire morte programmée par le fabricant
- PROM : Programmable Read Only Memory, mémoire morte programmable par l'utilisateur.
- EPROM : Erasable Programmable Read Only Memory

- **REPROM** : Reprogrammable Read Only Memory : Mémoires mortes programmable électriquement et, effaçable par ultraviolet, par l'utilisateur.
- **EEPROM** : Electrically Effaçable Programmable Read Only Memory
- **EAROM** : Electrically Altérable Read Only Memory : Mémoires mortes programmable et effaçable électriquement par l'utilisateur.

On peut également classer dans cette catégorie les réseaux logiques programmables qui permettent de réaliser des combinaisons logiques à partir de variables d'entrées.

- **PLA** : Programmable Logic Array, réseau logique programmable par l'utilisateur.

Mémoires à lecture et à écriture

- **Flash (NAND ou NOR)** : conserve les données enregistrées même lorsqu'elle n'est plus alimentée électriquement (non volatile).
- **RAM** : Random Access Memory, mémoire vive, deux types :
 - **RAM statique** : l'information est maintenue tant que l'alimentation est présente.
 - **RAM dynamique** : nécessite une opération de rafraîchissement (réécriture) pour maintenir l'information présente.

Les **DDR4 SDRAM** sont les RAM utilisées dans les ordinateurs actuellement (**Double Data Rate Synchronous Dynamic Random Access Memory** est un type de mémoire à circuit intégré)

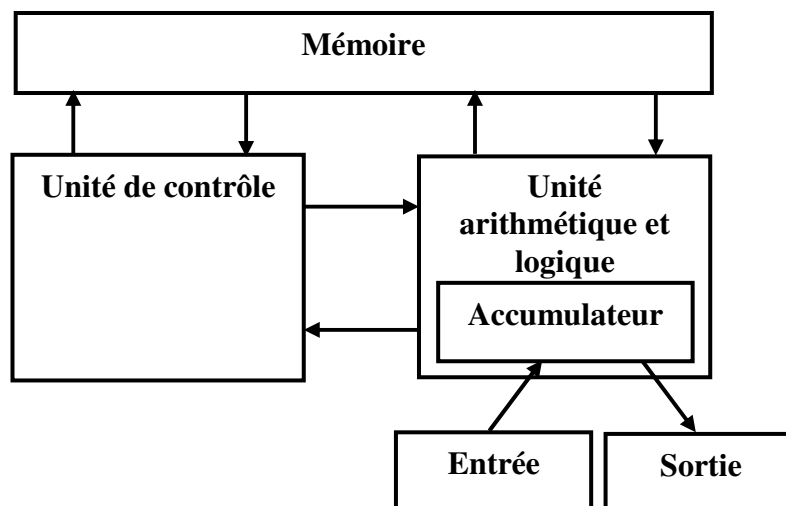
- **FIFO** : First In First Out, première information entrée première sortie, accès séquentiel (registre à décalage).
- **LIFO** : Last In First Out, dernière information entrée première sortie, accès séquentiel.

1.6. MODELE D'ARCHITECTURE SEQUENTIELLE DE VON NEUMANN

L'architecture générale des ordinateurs a été imaginée par John Von Neumann (1903-1957), mathématicien et physicien américano-hongrois en 1945.

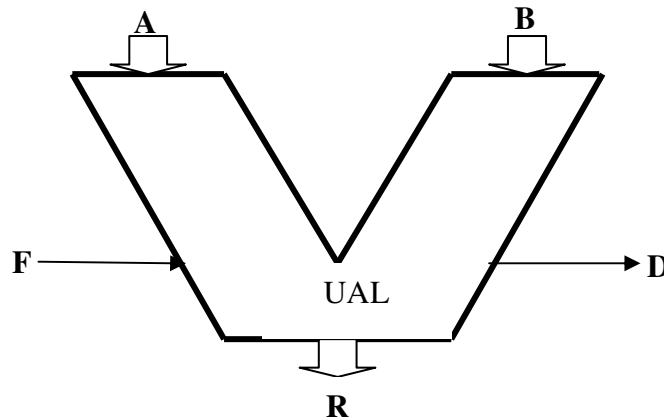
L'architecture de von Neumann décompose l'ordinateur en 4 parties distinctes :

- l'unité arithmétique et logique (UAL ou ALU en anglais) ou unité de traitement : son rôle est d'effectuer les opérations de base ;
- l'unité de contrôle, chargée du « séquençage » des opérations ;
- la mémoire qui contient à la fois les données et le programme qui indiquera à l'unité de contrôle quels sont les calculs à faire sur ces données. La mémoire se divise entre mémoire volatile (programmes et données en cours de fonctionnement) et mémoire permanente (programmes et données de base de la machine) ;
- les dispositifs d'entrée-sortie, qui permettent de communiquer avec le monde extérieur.



1.6.1. UAL

L'unité arithmétique et logique est le cœur de l'ordinateur. On la représente habituellement par ce schéma (en forme de V) :



- A et B sont les opérandes (entrées)
- R est le résultat (sortie)
- F est une opération binaire (opération à effectuer)
- D est un drapeau indiquant un résultat secondaire de la fonction (signe, retenue, erreur, etc.)

1.6.2. MEMOIRE

Les mémoires sont des « cases » contenant des données binaires. L'emplacement de chaque « case » est défini par son adresse.

1.6.3. UNITE DE CONTROLE

C'est elle qui donne les ordres à toutes les autres parties de l'ordinateur. Elle possède principalement quatre registres :

- CO (ou PC) : le compteur ordinal, qui contient l'adresse de la prochaine instruction à exécuter;
- RI (ou CIR): le registre qui contient l'instruction en cours;
- AC : l'accumulateur chargé de stocker des opérandes intermédiaires (opérande : argument)
- CC : le code condition pour les instructions de rupture conditionnelle

Il y a trois possibilités pour accéder à un opérande (valeur sur laquelle va être opérée une opération) :

- Il est directement saisi
- Il est dans un registre : il faut alors indiquer le nom de ce registre
- Il est dans la mémoire vive : il faut indiquer son adresse.

Il existe plusieurs langages assembleurs. Il faut être capable de s'adapter d'un langage machine à l'autre.

En langage assembleur, on peut rencontrer des calculs simples (addition, soustraction, multiplication), des affectations (placer telle valeur de la mémoire vive dans tel registre et vice versa) et enfin des instructions de saut qui permettent de coder des instructions conditionnelles par exemple.

1.7. SIMULATEUR DE CPU

Il existe autant de langages machines que de types de processeurs. Les commandes de base sont assez faciles à transposer d'un langage à l'autre.

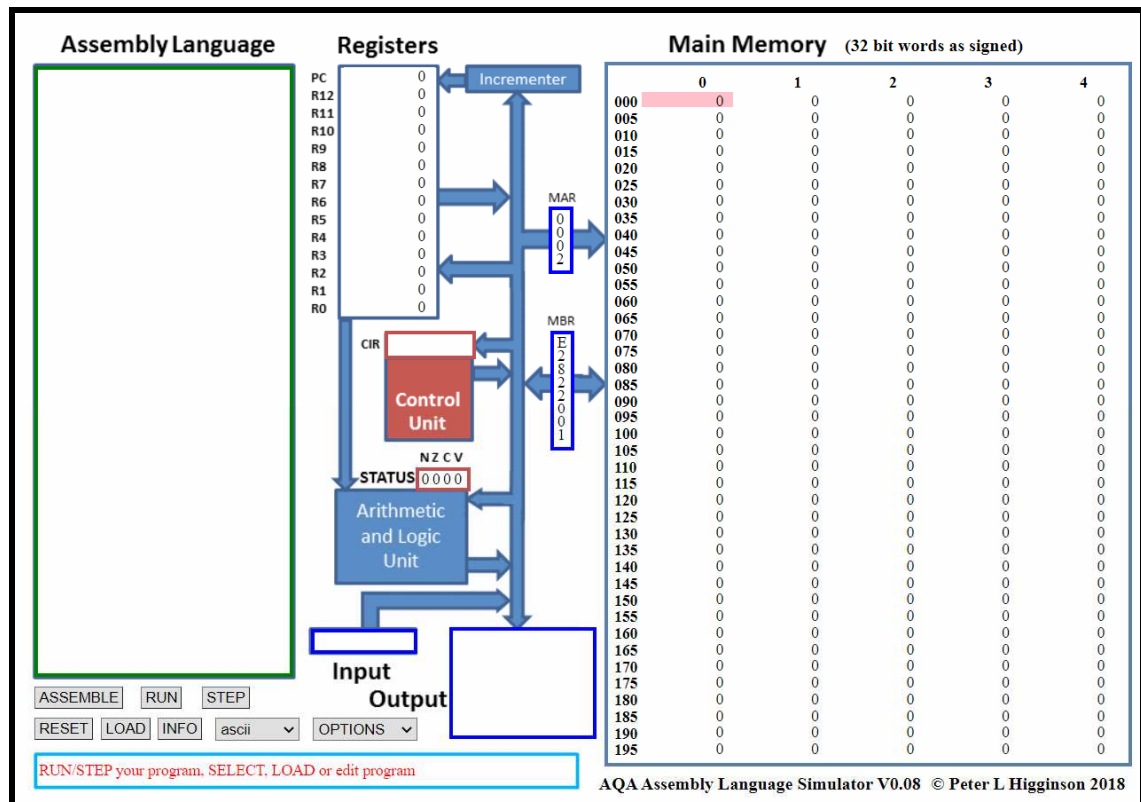
Voici le lien pour un simulateur de microprocesseur réalisé par Peter Higginson :

<http://www.peterhigginson.co.uk/AQA/>

Ce simulateur permet de tester et de visualiser l'exécution d'un programme écrit en langage assembleur.

Sur le schéma suivant, repérez les différentes zones suivantes :

- Programme assembleur
- Registres
- Compteur ordinal PC
- Registre qui contient l'instruction en cours CIR
- Mémoire
- Entrée
- Sortie



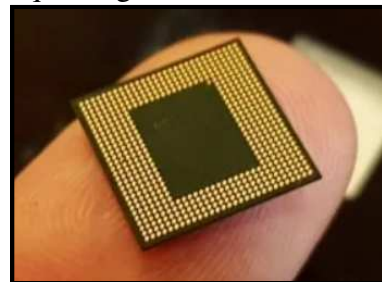
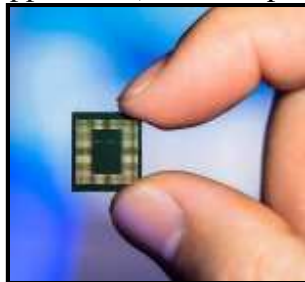
Liste d'instructions :

- LDR Rd, <adresse mémoire> Charge la valeur enregistrée dans l'<adresse mémoire> dans le registre d
- STR Rd, <adresse mémoire> Enregistre la valeur du registre d dans la mémoire spécifiée par <adresse mémoire>
- ADD Rd, Rn, <opérande 2> ajoute la valeur spécifiée par <opérande 2> à la valeur du registre n et enregistre le résultat dans le registre d
- SUB Rd, Rn, <opérande 2> Soustrait la valeur de <opérande 2> à la valeur du registre n et enregistre le résultat dans le registre d
- MOV Rd, <opérande 2> Copie la valeur <opérande 2> dans le registre d

- CMP Rn, < opérande 2 > Compare la valeur de registre n avec la valeur de <opérande 2>.
 - B <label> Branchement incondtionnel jusqu'à la positon <label> dans le programme.
 - B <condition> <label> Branchement conditionnel vers la position <label> dans le programme si la dernière comparaison remplit le critère spécifié par <condition>. Les valeurs possibles sont : EQ:égal à ; NE: n'est pas égal à; GT:Plus grand que; LT: Moins grand que.
 - AND Rd, Rn, < opérande 2 > Effectue l'opération bit à bit logique AND (ET) entre la valeur du registre n et la valeur < opérande 2 > et enregistre le résultat dans le registre d.
 - ORR Rd, Rn, < opérande 2 > Effectue l'opération bit à bit logique OR (OU) entre la valeur du registre n et la valeur < opérande 2 > et enregistre le résultat dans le registre d.
 - EOR Rd, Rn, < opérande 2 > Effectue l'opération bit à bit logique XOR(OU exclusif) entre la valeur du registre n et la valeur < opérande 2 > et enregistre le résultat dans le registre d.
 - MVN Rd, < opérande 2 > Effectue l'opération bit à bit logique NOT (NON) sur la valeur < opérande 2 > et enregistre le résultat dans le registre d.
 - LSL Rd, Rn, < opérande 2 > Décale de < opérande 2 > bit(s) vers la gauche la valeur du registre n et enregistre le résultat dans le registre d
 - LSR Rd, Rn, < opérande 2 > Décale de < opérande 2 > bit(s) vers la droite la valeur du registre n et enregistre le résultat dans le registre d
 - HALT Arrête l'exécution du programme.
 - < opérande 2 > peut être #nnn (c'est à dire un nombre, exemple #42) ou bien Rm (c'est à dire le registre m , par exemple R1 est le registre numéro 1)
 - Les registres vont de R0 à R12.
- Compléments data, INP et OUT :
- la pseudo instruction DAT vous permet de mettre un nombre dans la mémoire en utilisant l'assembleur. Un label peut aussi être pris comme donnée.
 - INP Rd,2 lit un nombre et l'enregistre dans le registre d.
 - OUT Rd,4 retourne en sortie le nombre du registre d
- Pour OUT, vous pouvez retourner des nombres signés (paramètre 4), des nombres non signés (paramètre 5), des hexadécimaux (paramètre 6) ou des caractères (paramètre 7). Vous pouvez entrer des hexa comme 0xnnn partout où un nombre est attendu.

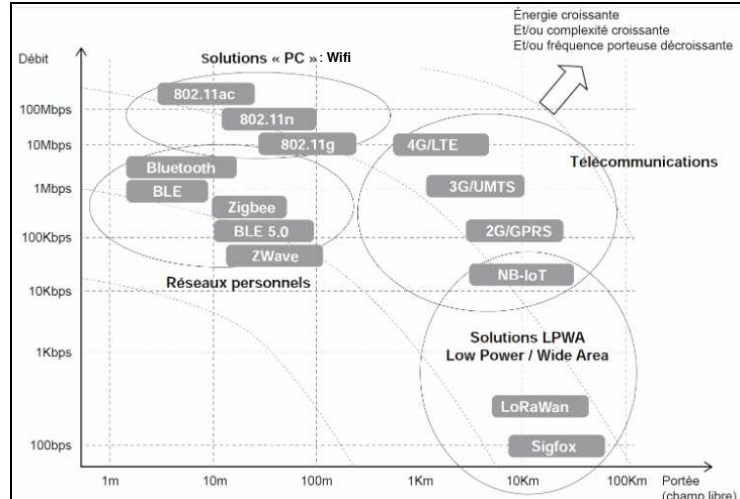
2. SYSTEMES SUR PUCE

La réduction de taille des éléments des circuits électroniques a conduit à la naissance de systèmes sur puce (**SoC : System on a Chip**) qui regroupent dans un seul circuit nombre de fonctions habituellement effectuées par des circuits séparés assemblés sur une carte électronique. Un tel système sur puce est conçu et mis au point de façon logicielle, ses « briques électroniques » sont accessibles par des API (API : ensemble de définitions et de protocoles qui facilite la création et l'intégration de logiciels d'applications. **API : Application Programming Interface** », interface de programmation d'application), comme pour les bibliothèques logicielles.



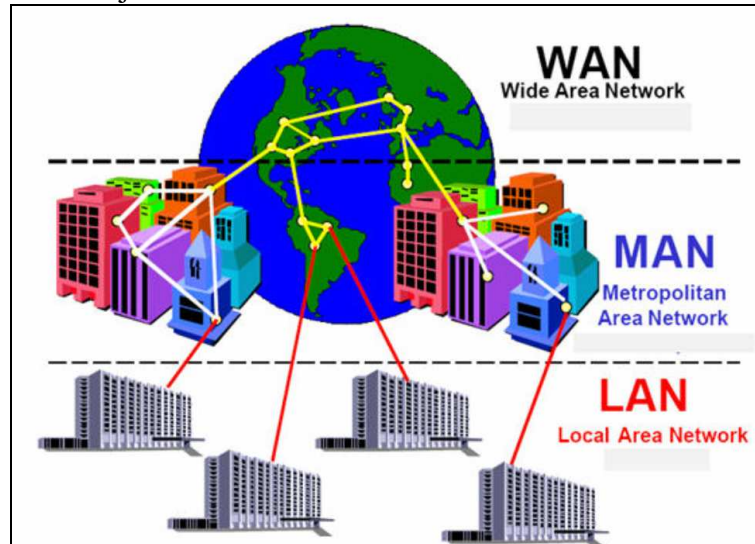
Un SoC comprend à la fois:

- un processeur central à un ou plusieurs cœurs de calcul (CPU)
- un processeur graphique (GPU)
- de la mémoire vive (RAM, Flash, ...)
- de la mémoire morte / statique (ROM, EPROM, ...)
- des modules de communications (Bluetooth, Wi-Fi, 2G/3G/4G/5G, LoRaWAN, ...)



www.elektormagazine.fr et <https://redmine.laas.fr/projects/lora/wiki>

LoRaWAN : protocole de télécommunication permettant la communication à bas débit, par radio, d'objets à faible consommation électrique communiquant selon la technologie LoRa et connectés à l'Internet via des passerelles, participant ainsi à l'Internet des objets.



<https://www.iifa.fr/reseaux>

LAN (Local Area Network : réseau local, échelle d'un bâtiment avec Wi-Fi et câble Ethernet) , MAN (Metropolitan Area Network : réseau métropolitain, échelle d'une ville avec modem et fils) et WAN (Wide Area Network : réseau à grande distance, échelle d'un pays ou un groupe de pays avec fibre optique et satellites. Ce sont les trois types de réseau conçus pour fonctionner sur la zone qu'ils couvrent. L'une des différences majeures est la zone géographique qu'ils couvrent, à savoir LAN couvre la plus petite zone, MAN couvre une zone plus grande que LAN et WAN couvre une zone plus grande que LAN et MAN), ...)

- des capteurs nécessaires au fonctionnement d'un smartphone ou d'un objet connecté (capteur gyroscopique , accéléromètre, capteur de lumière, microphone, ...)
- ...

En clair, le système sur une puce comprend tous les éléments essentiels d'un ordinateur comprimé dans une forme réduite. Son faible encombrement, son caractère complet et sa faible consommation d'énergie permettant de s'affranchir d'un système de refroidissement actif (ventilateur) en font un circuit intégré idéal pour les applications mobiles, notamment l'IoT. L'IoT (**I**nternet **o**f **T**hings) décrit le réseau de terminaux physiques, les appareils, qui intègrent des capteurs, des softwares et d'autres technologies en vue de se connecter à d'autres terminaux et systèmes sur Internet et d'échanger des données avec eux. Ces terminaux peuvent aussi bien être de simples appareils domestiques (montre connecté, téléphone, station météorologique, ...) que des outils industriels d'une grande complexité (dans ce cas on parle de IIot : **I**ndustrial **I**nternet **o**f **T**hings (production industrielle – auto surveillance de la chaîne de production, chaîne logistiques – gestion des stocks et des commandes, secteur de la santé – surveillance des patients à distance, commerce / marketing – ajustement de l'aspect des façades des magasins et des promotions en fonction des goûts des clients)).

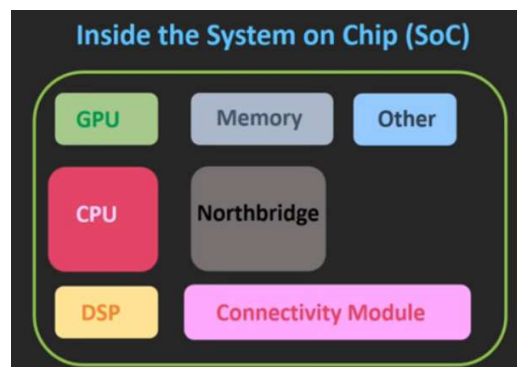
Les principaux concepteurs de « system on a chip » sont :

- AMD ;
- Apple ;
- Broadcom ;
- Intel ;
- Nvidia ;
- Qualcomm ;
- Samsung ;
- Texas Instrument.

Les SOC sont basés soit sur une architecture x86 (Intel uniquement) soit sur une architecture ARM, qui est la plus répandue.

2.1. SCHEMA D'UN SYSTEMES SUR PUCE

Visualisez la vidéo en anglais de présentation des SOC (activez si besoin les sous-titres et les traductions en français grâce aux paramètres, vous pouvez également si besoin ralentir la vitesse de lecture) : <https://www.youtube.com/watch?v=FUhCrWoNA2c>



Complétez les acronymes suivants et donnez leurs rôles :

- CPU :

- GPU :

- DSP :

- Mémoire :

- Autres :

- Northbridge :

- Modules de communications :

- Contrôleur DMA :

L'accès direct à la mémoire (Direct Memory Access – DMA) est un dispositif permettant le transfert de donnée entre un périphérique (lent) et la mémoire principale de la machine (rapide), sans intervention du microprocesseur. Ainsi, le microprocesseur n'est pas obligé d'attendre chaque donnée.

- NPU :

Circuit d'accélération de réseaux de neurones (Neural Processing Unit) c'est un accélérateur d'intelligence artificielle réalisé par une catégorie de microprocesseur ou de systèmes de calculs conçu pour accélérer un réseau de neurones artificiels, accélérer des algorithmes de vision industrielle et d'apprentissage automatique (deep learning : apprentissage) pour la robotique, l'Internet des objets et autres tâches de calculs intensifs ou de contrôle de capteurs.

2.2. LES AVANTAGES ET LES INCONVENIENTS D'UN SYSTEMES SUR PUCE

Outre leur taille miniaturisée bien adaptée aux terminaux nomades (smartphones, tablettes, montres connectées, ..), les SoC offrent d'autres **avantages** par rapport aux systèmes « classiques » rencontrés dans les ordinateurs :

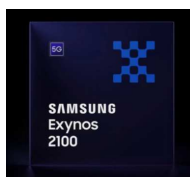
- Les SoC sont conçus pour **consommer** beaucoup **moins d'énergie** qu'un système classique à puissance équivalente de calculs.
- Cette consommation réduite d'énergie permet dans la plupart des cas de **s'affranchir** de la présence **d'un système de refroidissement actif** comme les ventilateurs ou de type « watercooling ». Un système équipé de SoC est donc **silencieux** car il chauffe relativement peu.
- Les distances étant très faibles (entre par exemple le CPU et la mémoire) les données circulent beaucoup plus vite, ce qui permet d'**améliorer les performances**. Dans les systèmes « classiques » les BUS chargés d'acheminer les données affaiblissent souvent les performances du système à cause de leur vitesse limitée de circulation des données.

Le principal **inconvénient** d'un SoC est que là où un ordinateur équipé d'une carte mère permet de faire évoluer les composants individuellement, l'extrême intégration du SoC présente l'inconvénient de **n'autoriser aucune mise à jour** possible du **matériel**. La **durée de vie** d'un Soc **dépend de son composant le plus faible** qui ne pourra ni évoluer, ni être réparé ou remplacé.

2.3. LES SYSTEMES SUR PUCE DANS LES SMARTPHONES

Les nouveaux procédés de gravure des semi-conducteurs CMOS telle que la lithographie extrême ultraviolette, ont permis de réduire significativement la taille des composants électroniques constituant les SoC. Ainsi, on dispose aujourd'hui de la même puissance dans un smartphone que celle embarquée dans un ordinateur il y a quelques années de cela. Ceci s'est cependant fait au prix d'une complexité technologique croissante. L'actuelle génération de SoC est gravée en 7 nm ($1 \text{ nm} = 10^{-9} \text{ m}$) depuis juillet 2019. Le SoC A13 Bionic d'Apple est par exemple composé de 8,5 milliards de transistors répartis sur une surface de $98,48 \text{ mm}^2$. La génération suivante gravée en 5 nm a vu le jour à partir de 2021. En novembre 2020, Samsung Electronics présentait son premier processeur 5 nm, l'Exynos 1080. Il était suivi en janvier 2021 par l'Exynos 2100, également 5 nm, qui équipe notamment les Galaxy S21 :

Nom du SoC	Gravure	Jeu d'instruction du CPU	CPU	GPU	Modèle smartphone
SAMSUNG Exynos 990	7nm	ARM v8.4	8 cœurs 64 bits	Mali-G77 MP11	Samsung Galaxy S20
APPLE A13-Bionic	7nm	ARM v8.3-A	6 cœurs 64 bits	Appledesigned 4 cores	Apple iPhone 11
QUALCOMM Snapdragon 865	7nm	ARM v8.2-A	8 cœurs 64 bits	Qualcomm Adreno 650	Xiaomi Mi 10
HISILICON Kirin 990	7nm	ARM v8.2-A	8 cœurs 64 bits	Mali-G76 MP16	Huawei P40
SAMSUNG Exynos 1080	5nm	ARM v8-A	8 cœurs 64 bits	Mali-G78 MP14	Vivo X60
SAMSUNG Exynos 2100	5nm	ARM v8-A	8 cœurs 64 bits	Mali-G78 MP14	Samsung Galaxy S21



2.4. LES INFORMATIONS IMPORTANTES SUR LES SYSTEMES SUR PUCE

Alors qu'un processeur « classique » est conçu pour effectuer les tâches qui lui sont confiées avec les meilleures performances possibles, le processeur d'un SoC est lui conçu pour être **le plus efficace** possible (efficace énergétiquement). Le but est qu'il **utilise le moins d'énergie possible** et que cette énergie (provenant de la batterie) soit **la plus rentable possible**. Ce principe s'applique également aux ordinateurs portables, dans lesquels on peut trouver des processeurs mobiles ou des SoC.

Afin d'obtenir de bonnes performances en ménageant la consommation d'un processeur, il est possible de jouer sur plusieurs facteurs. La **fréquence** du processeur, **le type de cœur** au sein du processeur, ainsi que **le procédé de gravure**.

2.4.1. LES FREQUENCES DES SYSTEMES SUR PUCE

La fréquence de fonctionnement est un facteur qui influe sur la consommation d'un processeur. Trop la baisser a de mauvaises conséquences sur ses performances du système. On trouve actuellement des systèmes sur puce ayant des fréquences de fonctionnement d'environ 1,3 à 3 GHz.

Donnez la fréquence de fonctionnement de :

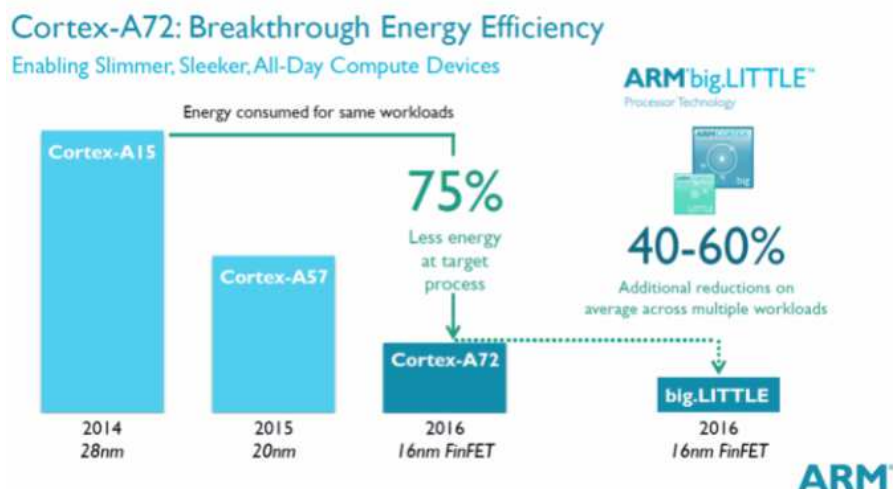
- Exynos 990 : _____
- Exynos 1080 : _____
- Exynos 2100 : _____
- A13-Bionic : _____
- Snapdragon 865 : _____
- Kirin 990 : _____

2.4.2. LE NOMBRE DE CŒURS DES SYSTEMES SUR PUCE

Le nombre de cœurs dans un SoC est également un facteur important qui influe sur la consommation et les performances du système.

2.4.3. LA GRAVURE DES SYSTEMES SUR PUCE

Les procédés de gravure sont également un facteur crucial pour que les SoC aient de bonnes performances et une faible consommation. Ils permettent d'améliorer les performances (augmentation du nombre de transistors) tout en limitant l'augmentation de la taille de la puce et la consommation énergétique.

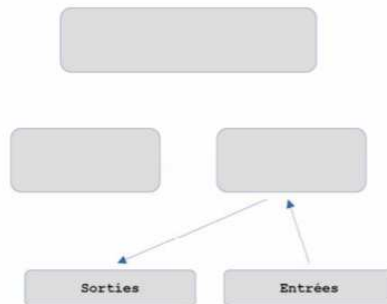


2.5. LES SYSTEMES SUR PUCE A L'EPREUVE ECRITE**2.5.1. SUJET 1, EXERCICE 4****Partie C : Architectures matérielles****Architecture Von Neumann**

« L'architecture dite architecture de Von Neumann est un modèle pour un ordinateur qui utilise une structure de stockage unique pour conserver à la fois les instructions et les données demandées ou produites par le calcul. De telles machines sont aussi connues sous le nom d'ordinateur à programme enregistré. » *source : Wikipédia*

Elle décompose l'ordinateur en 4 éléments : l'unité de contrôle (appelé aussi unité de commande), l'unité arithmétique et logique (UAL), la mémoire et les entrées-sorties. Les deux premiers éléments sont rassemblés dans le processeur (CPU en anglais pour Control Processing Unit).

1. Recopier et compléter le schéma de cette architecture ci-dessous en faisant apparaître les communications entre les différents éléments.

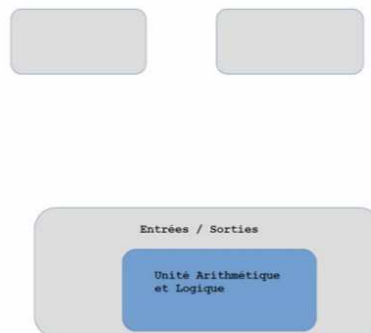


2. Dans quel(s) élément(s) sont situés le « compteur de programme » (CP ou IP en anglais pour Instruction Pointer) et le « registre d'instruction » (RI ou IR en anglais pour Instruction Register). Préciser leurs rôles.

Architecture de Harvard

« L'architecture de type Harvard est une conception qui sépare physiquement la mémoire de données et la mémoire programme. L'accès à chacune des deux mémoires s'effectue via deux bus distincts. [...] L'architecture Harvard est souvent utilisée dans les processeurs numériques de signal (DSP) et les microcontrôleurs. » *source : Wikipédia*

3. Recopier et compléter le schéma de cette architecture ci-dessous et faire apparaître les communications entre les différents éléments.



4. Expliquer ce qu'est une mémoire morte et une mémoire vive. Expliquer brièvement pourquoi, dans les microcontrôleurs, la mémoire programme est une mémoire morte.

Partie D : Système sur puce

« Un "système sur une puce", souvent désigné dans la littérature scientifique par le terme anglais "system on a chip" (d'où son abréviation SoC), est un système complet embarqué sur une seule puce ("circuit intégré"), pouvant comprendre de la mémoire, un ou plusieurs microprocesseurs, des périphériques d'interface, ou tout autre composant nécessaire à la réalisation de la fonction attendue. » *source : Wikipédia*

5. Citer un des avantages d'avoir plusieurs processeurs.
6. Expliquer pourquoi les systèmes sur puces intègrent en général des bus ayant des vitesses de transmission différentes.
7. Citer un des avantages d'un circuit imprimé de petite taille.
8. Citer un des inconvénients de cette miniaturisation.

2.5.2. SUJET 2, EXERCICE 2

Cet exercice porte sur les notions de routage, de processus et de systèmes sur puces.

Un constructeur automobile utilise des ordinateurs pour la conception de ses véhicules. Ceux-ci sont munis d'un système d'exploitation ainsi que de nombreuses applications parmi lesquelles on peut citer :

- un logiciel de traitement de texte ;
- un tableur ;
- un logiciel de Conception Assistée par Ordinateur (CAO) ;
- un système de gestion de base de donnée (SGBD).

Chaque ordinateur est équipé des périphériques classiques : clavier, souris, écran et est relié à une imprimante réseau.

1. Ce constructeur automobile intègre à ses véhicules des systèmes embarqués, comme par exemple un système de guidage par satellites (GPS), un système de freinage antiblocage (ABS) ...

Ces dispositifs utilisent des systèmes sur puces (SoC : Système on a Chip).

Citer deux avantages à utiliser ces systèmes sur puces plutôt qu'une architecture classique d'ordinateur.

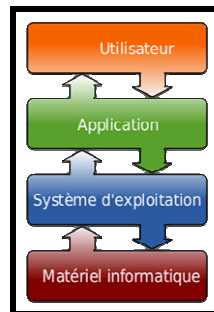
3. GESTION DES PROCESSUS ET DES RESSOURCES PAR UN SYSTEME D'EXPLOITATION

3.1. LES SYSTEMES D'EXPLOITATION (rappels de classe de première)

3.1.1. INTRODUCTION

En informatique, un système d'exploitation (souvent appelé OS de l'anglais Operating System : système d'exploitation) est un ensemble de programmes qui dirige l'utilisation des ressources d'un ordinateur par des logiciels applicatifs. Il reçoit des demandes d'utilisation des ressources de l'ordinateur de la part des logiciels applicatifs : demandes de ressources de stockage des mémoires (accès à la mémoire vive, aux disques durs, ressources de CPU (Central Processing Unit), demandes de ressources de communication vers des périphériques (pour parfois demander des ressources de calcul au GPU (Graphics Processing Unit) ou tout autre carte d'extension) ou demandes via le réseau. Le système d'exploitation gère les demandes ainsi que les ressources nécessaires, évitant les interférences entre les logiciels.

Un système d'exploitation représente l'ensemble des programmes qui pilote les différents composants (disque dur, écran, processeur, mémoire etc...) de l'appareil informatique et lui permet donc de fonctionner. Il permet donc de faire l'interface entre l'utilisateur et le matériel informatique.



Le système d'exploitation est le deuxième programme exécuté après le programme d'amorçage (bootloader en anglais) lors de la mise en marche de l'ordinateur.

Il existe sur le marché des dizaines de systèmes d'exploitation différents, très souvent livrés avec l'appareil informatique. C'est le cas de **Windows** (créé par Microsoft en 1985), **Mac OS** (créé en 1998 par Apple sur un noyau Unix (Unix créé en 1969 par Ken Thompson et Dennis Ritchie chez AT&T Bell)), **GNU/Linux** pour lequel il existe de nombreuses distributions **libre** et **open source** (GNU (GNU is Not Unix) créé en 1983 en s'inspirant d'Unix par Richard Stallman (fondateur de l'association Free Software Foundation : liberté d'exécution, de modification, de redistribution et d'amélioration ; la licence GPL est le cadre juridique des logiciels du projet GNU qui reprend les quatre libertés fondamentales et impose pour la liberté d'amélioration que les versions modifiées d'un logiciel sous licence GPL ne peuvent être que redistribuées sous cette même licence) ; Linux créé en 1991 par Linus Torvalds en s'inspirant du noyau Unix), **Android** (créé en 2007 sur un noyau Linux par Google) ou **iOS** (iphone OS, créé en 2008 par Apple sur un noyau Unix). Les fonctionnalités offertes diffèrent d'un système à l'autre et sont typiquement en rapport avec l'exécution des programmes, l'utilisation de la mémoire centrale ou des périphériques, la manipulation des systèmes de fichiers, la communication, ou la détection et la gestion d'erreurs.

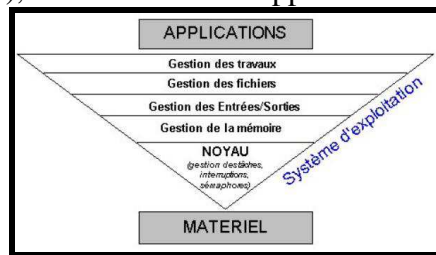
Actuellement, les deux familles de systèmes d'exploitation les plus populaires sont **UNIX** (dont Mac OS, GNU/Linux, iOS et Android) et **WINDOWS**. Ce dernier détient un quasi-monopole sur les ordinateurs personnels avec près de 90 % de part de marché.

3.1.2. FONCTIONNALITES ET CONSTITUTION DES SYSTEMES D'EXPLOITATION

Le principal rôle du système d'exploitation est alors de gommer les différences entre les différentes architectures informatiques, et d'organiser l'utilisation des ressources de façon rationnelle :

- **Utilisation des périphériques** : chaque périphérique a ses propres instructions, avec lesquelles il peut être manipulé. Le système d'exploitation en tient compte. Il permet à l'utilisateur de manipuler le périphérique par de simples demandes de lecture ou d'écriture, lui évitant la perte de temps pour traduire les opérations en instructions propres au périphérique.
- **Accès aux fichiers** : en plus des instructions propres à chaque périphérique (ROM, RAM, disque dur, lecteur de CD-ROM ou DVD-ROM, SSD, clé USB, ...), le système d'exploitation tient compte du format propre de chaque support servant au stockage des données. Il offre également des mécanismes de protection permettant de contrôler quel utilisateur peut manipuler quel fichier.
- **Accès aux ressources** : une des fonctions du système d'exploitation est de protéger les ressources contre l'utilisation par des personnes non autorisées, et de résoudre les conflits lorsque deux utilisateurs demandent simultanément la même ressource.
- **Détection et récupération en cas d'erreur** : lorsqu'une erreur survient, qu'elle soit matérielle ou logicielle, le système d'exploitation traite l'erreur en adoucissant son impact sur le système informatique. Il peut tenter de réitérer l'opération, arrêter l'exécution du programme fautif, ou signaler le problème à l'utilisateur.
- **Contrôle** : un système d'exploitation peut tenir des statistiques d'utilisation des ressources, surveiller la performance et les temps de réponse.

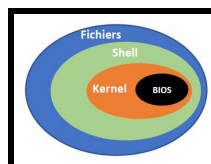
Le système d'exploitation (OS), est chargé d'assurer la liaison entre les ressources matérielles (hardware), l'utilisateur et les applications.



La palette des services offerts et la manière de s'en servir diffèrent d'un système d'exploitation à l'autre.

Un système d'exploitation se compose de 3 éléments principaux :

- Un **noyau (kernel)** qui représente les fonctions fondamentales du système informatique (à savoir la gestion des périphériques, de la mémoire, ...) . Dans le noyau, il y a le **BIOS** (Basic Input Output System : système élémentaire d'entrée/sortie), un ensemble de fonctions, contenu dans la mémoire morte (ROM) de la carte mère d'un ordinateur, lui permettant d'effectuer des opérations de base, lors de sa mise sous tension. Par exemple l'identification des périphériques d'entrée/sortie connectés et la lecture d'un secteur sur un disque, un CD ou une partie d'une clé USB.
- Un **interpréteur de commande** ou **Shell** (coquille en opposition au noyau) permet d'assurer la communication avec le système d'exploitation par le biais d'un langage de commande pour permettre à l'utilisateur d'utiliser les ressources matérielles du système informatique sans connaître leur caractéristique.
- Un **système de fichiers** permettant d'enregistrer les données dans une arborescence.



3.1.3. RESUME SUR LES OS (SYSTEMES D'EXPLOITATION)

Visualisez la vidéo de présentation des OS (si besoin, grâce aux paramètres, vous pouvez ralentir la vitesse de lecture) : <https://www.youtube.com/watch?v=AcZ87MTiXr4>

Le système d'exploitation d'un ordinateur est chargé d'assurer les fonctionnalités de communication et d'interfaçage avec l'utilisateur. Un OS est un logiciel dont le principal domaine d'intervention est la gestion de toutes les ressources de l'ordinateur comme:

- Le chargement et le lancement des programmes
- La gestion des processeurs, de la mémoire, des périphériques
- La gestion des processus (programmes en cours d'exécution) et des fichiers
- La protection contre les erreurs et la détection des erreurs
- ...

C'est donc une composante logicielle très importante.

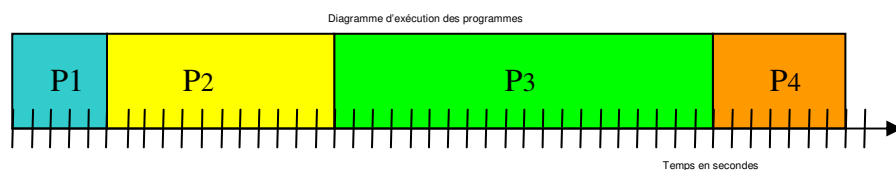
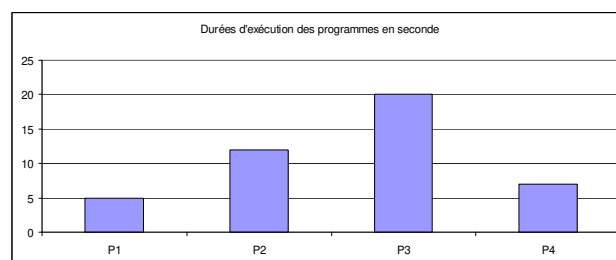
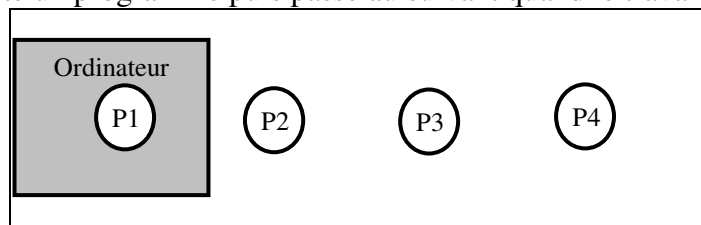
Dans un ordinateur, les logiciels sont divisés en deux catégories :

- Les programmes système qui font fonctionner l'ordinateur : le système d'exploitation et les utilitaires (compilateurs, éditeurs, interpréteurs de commandes, ...)
- Les programmes d'application qui résolvent les problèmes spécifiques des utilisateurs.

3.1.4. PRINCIPAUX TYPES D'OS (SYSTEMES D'EXPLOITATION)

Un OS (système d'exploitation) est dit « **monoprogrammation** », lorsqu'un seul programme a accès à toutes les ressources de la machine pendant tout le temps que dure son exécution. Si plusieurs programmes doivent être exécutés, ils le seront les uns après les autres.

Supposons que pour un tel système, 4 programmes (P1, P2, P3 et P4) s'exécutent. En réalité, l'OS exécute un programme puis passe au suivant quand le travail est terminé :



Pour atteindre la fin du programme P1, il faudra attendre 5s.

Pour atteindre la fin du programme P2, il faudra attendre 5s + 12s = 17s.

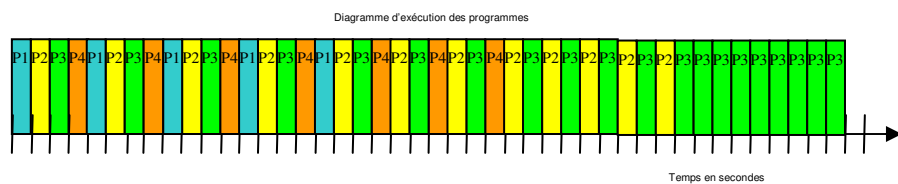
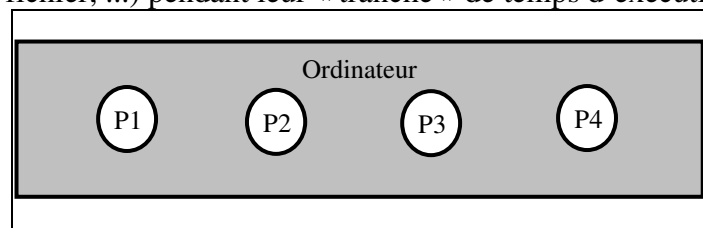
Pour atteindre la fin du programme P3, il faudra attendre $5s + 12s + 20s = 37s$.

Pour atteindre la fin du programme P1, il faudra attendre $5s + 12s + 20s + 7s = 44s$.

Même si les temps d'exécution des programmes sont faibles, les temps d'attente se cumulent.

Un OS est dit « **multiprogrammation** » lorsque plusieurs programmes peuvent s'exécuter en « même temps » sur la machine et se partagent les ressources de celle-ci pendant toute la durée de leur exécution. La notion de multiprogrammation est la capacité pour un système d'exécuter à la suite plusieurs activités sans l'intervention de l'utilisateur. Elle apparue dans les années 1960.

En reprenant l'exemple ci-dessus avec les programmes (P1, P2, P3, P4) à exécuter, il seront exécutés cycliquement par l'OS qui leur alloue les ressources nécessaires (disque, mémoire, fichier, ...) pendant leur « tranche » de temps d'exécution.



Dans un système multiprogrammation les « petits » programmes atteignent la fin de leur exécution plus rapidement que les « gros ». Actuellement, tous les OS sont issus de la multiprogrammation. Elle est apparue dans la famille Windows avec Win98.

QCM : Quel est le rôle du système d'exploitation (OS) ?

Réponses :

- Allumer le disque dur.
- Compiler et exécuter un programme.
- Partager la machine physique entre les différents programmes et bâtir une IHM accessible à l'utilisateur.
- Gérer les accès du disque dur.

3.2. LES PROCESSUS

3.2.1. LES PROGRAMMES

Un programme est un ensemble d'instructions permettant de faire réaliser certaines tâches à un système informatique. Il s'agit donc d'une implémentation en machine d'un algorithme (algorithme qui peut lui être vu comme l'idée abstraite de l'action à accomplir).

En réalité, derrière ce mot "programme" se trouve deux notions :

- Un programme binaire (ou langage machine) :
Il s'agit d'une suite de bits directement compréhensible par le processeur.

Exemple d'un ensemble de bits permettant de transférer le contenu d'une zone - mémoire dans l'un des registres de processeur :

En binaire : 0010 0010 0011 1001 0000 0000 0000 0000 0000 0010 0000

En hexadécimal : 2239 0000 0020

Il s'agit du jeu d'instructions 68000 de Motorola, on peut lire dans la notice que « 001 » en début d'instruction signifie qu'on veut déplacer des données en mémoire (dont on donne l'adresse) vers un registre.

La traduction mot à mot en utilisant quelques mots clés plutôt que les valeurs réelles des bits se nomme le langage d'assemblage ou assembleur (par abus de langage).

- Un programme source (ou code source) :

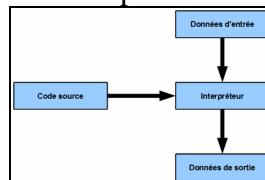
Il s'agit des instructions fournies dans un langage de programmation.

Ce code - source est compréhensible par un humain.

Ce code - source n'est pas compréhensible par le processeur d'un système informatique et doit être traduit en langage machine.

Voyons deux manières de gérer cette traduction :

- Les langages interprétés (Java, Python, ...). Dans ces langages, le code source (celui que vous écrivez) est interprété, par un logiciel qu'on appelle interpréteur. Celui-ci va utiliser le code source et les données d'entrée pour calculer les données de sortie :

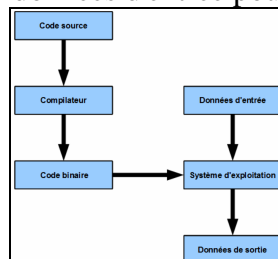


L'interprétation du code source est un processus « pas à pas » : l'interpréteur va exécuter les lignes du code une par une, en décidant à chaque étape ce qu'il devra faire ensuite.

Avantages : Dans un langage interprété, le code source pourra **fonctionner directement sur tous les ordinateurs**.

Inconvénients : **Lent** car le code source doit être interprété à chaque exécution.

- Les langages compilés (C, C++, Pascal, OCaml,...). Dans ces langages, le code source (celui que vous écrivez) est tout d'abord compilé, par un logiciel qu'on appelle compilateur, en code binaire qu'un humain ne peut pas lire mais qui est très facile à lire pour un ordinateur. C'est alors directement le système d'exploitation qui va utiliser le code binaire et les données d'entrée pour calculer les données de sortie :



Avantages : Dans un langage compilé, le programme est directement exécuté sur l'ordinateur, donc il sera en général **plus rapide** car il est inutile de recompiler à chaque exécution (La compilation n'est que nécessaire lors de la modification du code source).

Inconvénients : Après compilation, le **code machine** (code binaire) n'est **plus compréhensible** par l'humain.

3.2.2. LES PROCESSUS

3.2.2.1. NOTION DE CONTEXTE D'EXECUTION D'UN PROGRAMME

Lorsqu'un programme (qui a été traduit en instructions machines) s'exécute, le processeur central lui fournit toutes ses ressources (registres internes, place en mémoire centrale, données, code,...), on appelle cet ensemble de ressources mises à disposition d'un programme « **son contexte d'exécution** ».

3.2.2.2. PROCESSUS ET ORDONNANCEMENT

Un **processus** est l'image en mémoire centrale d'un programme s'exécutant avec son contexte d'exécution.

Un ordinateur possède un ou plusieurs processeurs, qui sont eux-mêmes constitués de plusieurs unités de calcul, les cœurs. C'est le système d'exploitation qui va donner à un processus l'accès à une unité de calcul, cela s'appelle **l'ordonnancement**. Les processus ne quitteront cette dernière que si :

- Le processus s'arrête, lorsqu'il est terminé.
- Le processus demande à partir, il n'est pas terminé et demandera à revenir plus tard pour poursuivre son exécution. C'est par exemple le cas d'une tâche cyclique.
- Le processus est en attente. C'est par exemple, le cas si il n'y a pas d'instance (exemplaire) disponible de la ressource demandée par le processus, il est mis en attente pour libérer la place sur l'unité de calcul.
- Le système choisit d'arrêter le processus. C'est par exemple, le cas lors de l'interblocage.

Lorsqu'une unité de calcul est libre, c'est le système d'exploitation qui va déterminer un nouveau processus à affecter à l'unité de calcul. Pour cela il existe plusieurs algorithmes d'ordonnancement :

- Le modèle **FIFO** : on exécute les processus dans l'ordre de leur apparition dans la file d'attente.
- Le modèle **SJF (Shortest Job First)** : on exécute en premier le « plus court processus en premier » de la file d'attente à l'unité de calcul.
- Le modèle **Round Robin** : (ou méthode du tourniquet) on exécute un bloc de chaque processus présent dans la file d'attente à tour de rôle, pendant un quantum de temps d'en général 20 à 30 ms. Si le processus n'est pas terminé, il repart en fin de liste d'attente.
- Il existe d'autres algorithmes d'ordonnancement, comme par exemple le modèle **Priorité**, où chaque processus dispose d'une valeur de priorité et on choisit le processus de plus forte priorité à chaque fois.

Afin de savoir si un algorithme est préférable pour un ensemble de processus voici quelques définitions :

- **Représentation de l'ordonnancement**

Réaliser l'ordonnancement d'une succession de processus c'est compléter le tableau de ce processus :

Processus	P1	P2	P3	...
Durée en quantum				
Date d'arrivée				
Temps de terminaison (option)				
Temps d'exécution				
Temps d'attente				

Le processeur exécute chaque processus pendant quelques dizaines ou centaines de millisecondes. Le temps d'allocation du processeur au processus est appelé **quantum**.

Le **temps ou la date d'arrivée** correspond au moment où le processus arrive dans la file d'attente.

La **durée du processus** ou durée d'exécution sur le coeur, correspond à la durée en quantum nécessaire à l'exécution du processus.

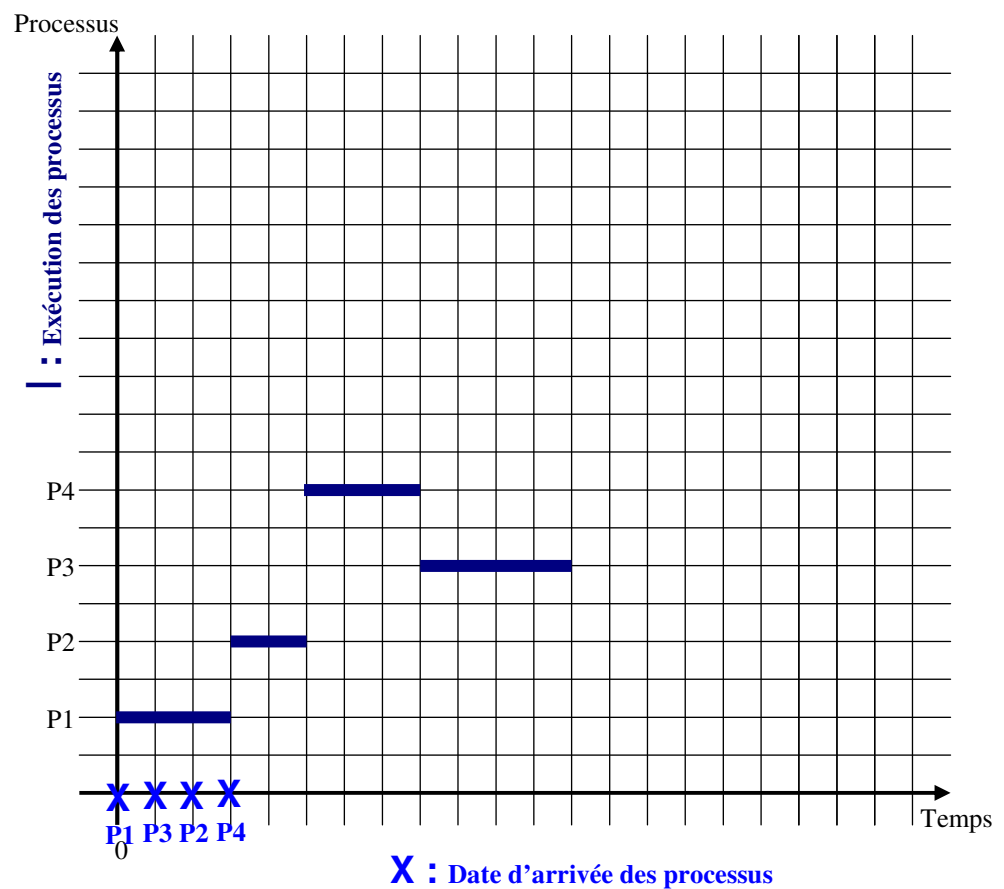
Le **temps d'attente** est le temps entre la date d'arrivée du processus et le début du processus.

Le **temps d'exécution** ou temps de séjour est le temps d'attente plus la durée du processus.

Le **temps de terminaison** est la durée écoulée entre le temps 0 et la fin du processus.

* Avec le modèle **SJF** (Shortest Job First : processus le plus court effectué en premier) :

Processus	P1	P2	P3	P4
Durée en quantum	3	2	4	3
Date d'arrivée	0	2	1	3
Temps d'attente				
Temps d'exécution				
Temps de terminaison				

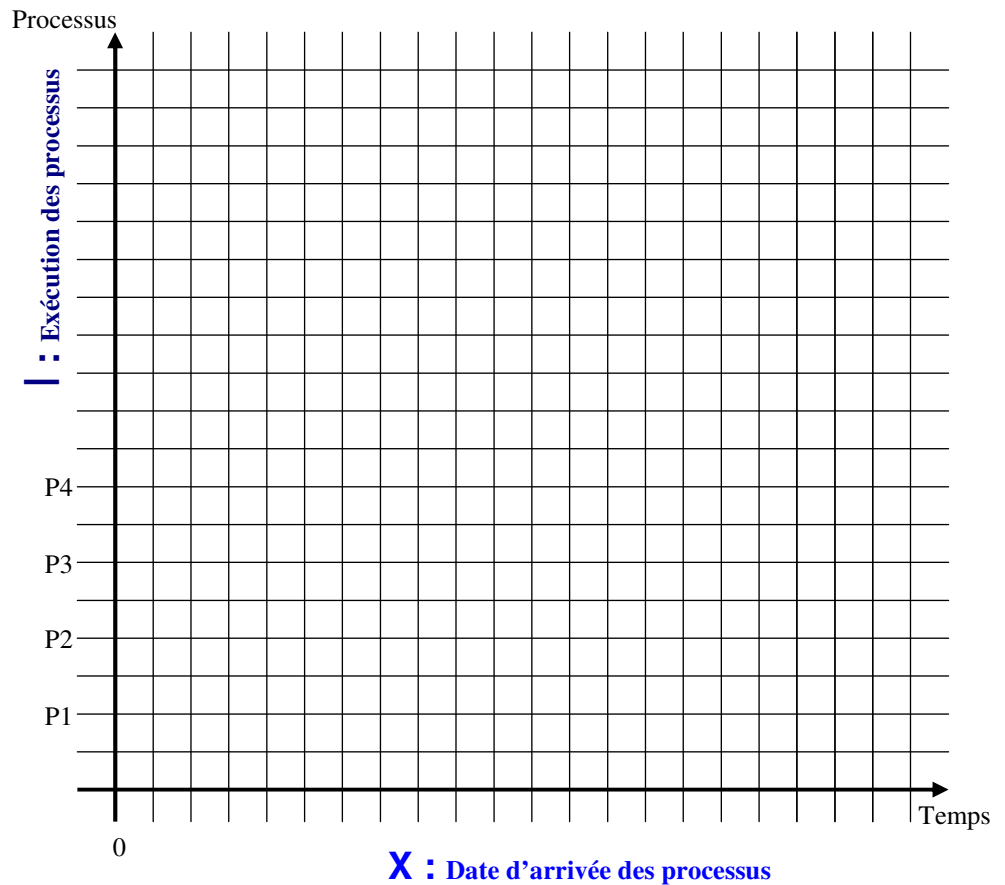


Complétez les temps d'attente, temps d'exécution et temps de terminaison du tableau ci-dessus, puis calculez les temps d'attente moyen et temps d'exécution moyen.

* Avec le modèle **FIFO** (First In First Out : processus arrivé en premier exécuter en premier) :

Processus	P1	P2	P3	P4
Durée en quantum	3	2	4	3
Date d'arrivée	0	2	1	3
Temps d'attente				
Temps d'exécution				
Temps de terminaison				

Complétez le schéma suivant :

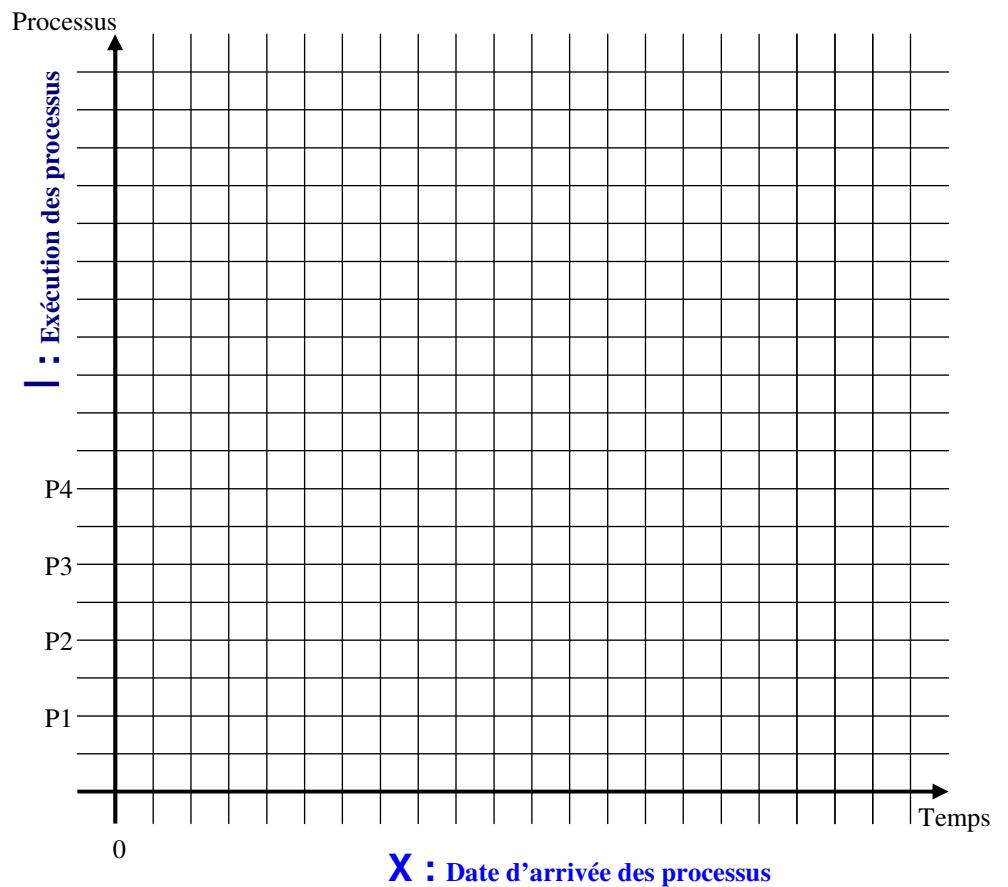


Complétez les temps d'attente, temps d'exécution et temps de terminaison du tableau ci-dessus, puis calculez les temps d'attente moyen et temps d'exécution moyen.

* Avec le modèle **Round Robin** : effectuer un quantum de chaque processus présent dans la file d'attente à tour de rôle, si le processus n'est pas terminé, il repart en fin de liste d'attente :

Processus	P1	P2	P3	P4
Durée en quantum	3	2	4	3
Date d'arrivée	0	2	1	3
Temps d'attente				
Temps d'exécution				
Temps de terminaison				

Complétez le schéma suivant :



Complétez les temps d'attente, temps d'exécution et temps de terminaison du tableau ci-dessus, puis calculez les temps d'attente moyen et temps d'exécution moyen.

Actuellement, la plupart des systèmes d'exploitation utilisent une évolution du modèle priorité, reposant sur les principes suivants :

- Chaque processus possède une priorité de base.
- Cette priorité augmente quand le processus est inactif et diminue quand il est actif (le taux de changement dépend de la priorité de base).
- Le système choisit parmi les processus de plus forte priorité.

Exercices :

* Représentez le schéma de l'ordonnancement des processus ci-dessous avec le modèle FIFO :

Processus	P1	P2	P3
Durée en quantum	8	5	10
Date d'arrivée	8	3	0

* Représentez le schéma de l'ordonnancement des processus ci-dessous avec le modèle SJF :

Processus	P1	P2	P3	P4
Durée en quantum	8	5	9	2
Date d'arrivée	4	0	3	7

* Représentez le schéma de l'ordonnancement des processus ci-dessous avec le modèle Round Robin :

Processus	P1	P2	P3
Durée en quantum	8	5	10
Date d'arrivée	1	0	3

* Représentez le schéma de l'ordonnancement des processus ci-dessous avec le modèle Round Robin et complétez le tableau suivant :

Processus RR	P1	P2	P3	P4	P5	P6	P7
Durée en quantum	8	5	9	2	6	8	4
Date d'arrivée	1	0	2	3	4	6	5
Temps d'attente							
Temps d'exécution							
Temps de terminaison							

* Représentez les schémas de l'ordonnancement des processus ci-dessous avec les modèles FIFO et SJF Robin et complétez les tableaux suivants :

Processus FIFO	P1	P2	P3	P4	P5	P6	P7
Durée en quantum	8	5	9	2	6	8	4
Date d'arrivée	1	0	2	3	4	6	5
Temps d'attente							
Temps d'exécution							
Temps de terminaison							

Processus SJF	P1	P2	P3	P4	P5	P6	P7
Durée en quantum	8	5	9	2	6	8	4
Date d'arrivée	1	0	2	3	4	6	5
Temps d'attente							
Temps d'exécution							
Temps de terminaison							

Parmi les 3 modèles (RR, SJF et FIFO), quel est le plus efficace pour l'exécution des processus ?

3.2.2.3. QUESTIONS

- Qu'est-ce qui limite concrètement le nombre de processus pouvant être lancé en même temps ?
- Qu'est-ce qu'un système d'exploitation multitâche ?
- Que va devoir faire un système d'exploitation multitâche lorsque plusieurs processus fonctionnent "en même temps" sur un ordinateur ne disposant que d'un seul microprocesseur (un coeur) ?
- Que se passe-t-il au démarrage de l'ordinateur ?

Premier démarrage : le chargeur d'amorçage. Lorsqu'on démarre un système informatique, il est conçu pour générer automatiquement un premier processus en allant chercher un programme dans une zone précise de la mémoire : le **chargeur d'amorçage** (bootloader). Ce processus va alors déclencher tous un tas d'autres processus : ceux permettant le démarrage du système d'exploitation par exemple.

Les démons (daemons) : Il s'agit d'une catégorie particulière de processus : des processus qui fonctionnent en boucle car ils surveillent spécifiquement certaines choses. Leurs noms finissent souvent par ...d. Exemples :

- **httpd** : le processus d'un serveur http
- **crond** : le processus du planificateur de tâches Cron (pour Chrono Table) de Linux (et Unix)

Une **application** peut être vue comme étant un ensemble de processus produisant un effet commun. La plupart des applications réelles ne sont pas modélisables comme un seul processus : on encapsule certaines de leurs fonctions dans des processus indépendants discutant entre eux en utilisant des signaux et de la mémoire partagée.

3.2.2.4. ETATS DU PROCESSUS ET FONCTIONNEMENT

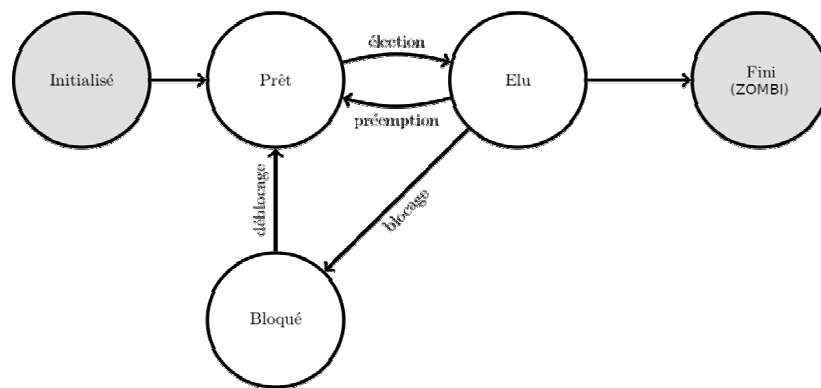
3.2.2.4.1. ETATS DU PROCESSUS, CYCLE DE VIE DU PROCESSUS

* Le programme est initialement en mémoire. Il est dans un premier temps « **initialisé** » : la place en mémoire vive est réservée pour y stocker le code (chargement des instructions, allocation de mémoires et des ressources).

Il existe 4 événements pour créer un processus :

- L'initialisation du système : au chargement du système il y a création automatique du processus racine père de tous les processus utilisateurs (id=0).
- Un processus peut lancer un autre processus, celui-ci en devient le parent, tandis que l'autre sera désigné comme processus fils. Le processus père ne se termine que lorsque tous ses fils sont terminés. On a donc une structure arborescente de processus).
- Une requête de l'utilisateur.
- Initiation d'un travail en traitement par lot (on exécute le même travail sur plusieurs entités à la fois, c'est par exemple très utilisé en photographie où on applique la même action sur plusieurs clichés).

Lors de sa création, un numéro unique d'identification est attribué à chaque processus, c'est l'**identifiant de processus ou PID** (Process IDentifier). Grâce à cet identifiant, nous pourrions appliquer différentes commandes sur le processus.



* Le processus passe alors automatiquement à l'état « **Prêt** » : il est prêt à faire exécuter sa prochaine instruction élémentaire au processeur. Il est mis en attente jusqu'à ce qu'on lui libère le processeur (dispatch de l'Ordonnanceur), il passera alors à l'état actif ou « **Elu** ».

* Le système d'exploitation va alors élire le prochain processus pouvant accéder au processeur : l'élection. Le processus passe à l'état « **Elu** » s'il remporte l'élection : ce sont ses instructions qui vont être traitées par le processeur.

Trois situations alors :

- Le processus n'a plus d'instructions à traiter mais les ressources allouées ne sont pas libérées : il passe à l'état « **Fini** » (zombi). En réalité, il existe 4 raisons pour qu'un processus se trouve dans l'état « **Fini** » (zombi) :
 - Arrêt normal : cet arrêt est volontaire et intervient lorsque le processus termine sa tâche.
 - Arrêt pour erreur : cet arrêt est volontaire, il fait suite à une erreur pour une instruction illégale.
 - Arrêt pour erreur fatale : cet arrêt est involontaire et intervient généralement lorsque les paramètres de l'exécution du processus sont mauvais.
 - Arrêt volontaire par un autre processus.

- Le processus est en attente d'une réponse, d'une ressource ou d'une entrée : il passe à l'état « **bloqué** ».
 - Le processeur a décidé de donner l'accès à un autre processus : il passe à l'état « **Prêt** ».
- * Le processus passe à l'état « **bloqué** ». Il pourra revenir à l'état « **Prêt** » lorsque le système aura reçu la réponse, la ressource ou l'entrée qu'il attend pour ce processus.

Quels sont les états pendant lesquels la mémoire vive réserve de la place au processeur ?

3.2.2.4.2. FONCTIONNEMENT DU PROCESSUS

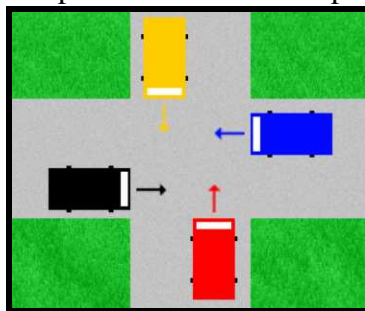
Un processus va trouver à sa disposition un grand nombre de ressources, comme la RAM, les disques durs, les supports amovibles (clés USB, ...), les fichiers, ...

Lorsqu'un processus est dans l'état actif ou élu, dans des conditions normales de fonctionnement, il ne peut utiliser une ressource qu'en suivant la séquence de trois étapes suivante : **Requête – Utilisation - Libération**

- La requête : le processus fait une demande pour utiliser la ressource. Si cette demande ne peut pas être satisfaite immédiatement, parce que la ressource n'est pas disponible, le processus demandeur se met en état attente (état « bloqué ») jusqu'à ce que la ressource devienne libre. Il devra ensuite attendre que le processus qui l'a remplacé soit terminé pour poursuivre son exécution.
- Utilisation : Le processus peut exploiter la ressource.
- Libération : Le processus libère la ressource qui devient disponible pour les autres processus éventuellement en attente.

3.2.2.4.3. NOTION D'INTERBLOCAGE (DEADLOCK)

Un **interblocage** est une situation assez commune dans la vie de tous les jours, notamment sur un carrefour avec priorité à droite : c'est une situation où personne ne peut plus agir à cause du comportement des autres protagonistes :

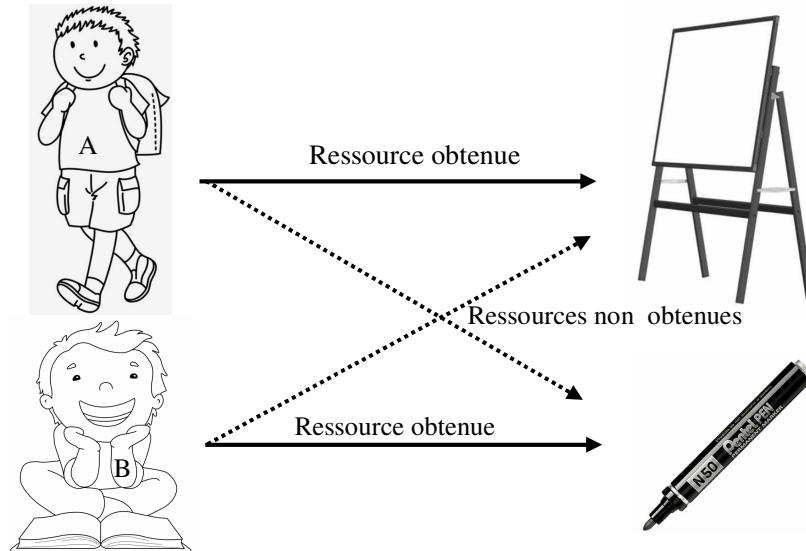


<http://le-permis.ch/theorie/5-priorite-droite-inverse.html>

En Informatique, cette situation peut se produire si plusieurs processus se bloquent les uns les autres. Comment alors que nous avons vu que l'Ordonnanceur parvenait à alterner ? Principalement à cause des accès aux ressources (écriture sur un fichier, accès à un périphérique n'autorisant pas les utilisateurs multiples...)

Un ensemble de processus est dans une situation d'interblocage si chaque processus de l'ensemble attend un événement qui ne peut être produit que par un autre processus de l'ensemble.

Exemple concret : On demande à deux élèves de dessiner une maison au tableau. On donne le tableau au premier et le feutre au second. Les deux élèves ne peuvent pas se prêter ou céder les ressources feutre et tableau. Ils vont se trouver dans une situation d'attente interminable puisque le premier élève attend que la ressource feutre soit libre et le second attend que la ressource tableau soit libre. Le dessin ne sera jamais fait.

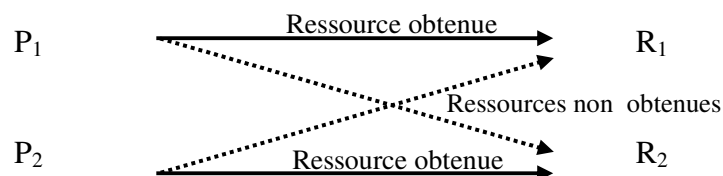


Exemple informatique : On prend deux processus P_1 et P_2 concurrents, qui ont besoin d'utiliser simultanément deux ressources R_1 et R_2 non partageables pour s'exécuter. Les processus demandent les ressources une par une.

Au départ, les deux ressources sont « libres ».

- L'Ordonnanceur passe le processus P_1 de l'état « prêt » à l'état « élu » : celui-ci demande alors la ressource R_1 . Celle-ci est libre, donc il l'obtient.
Le processus P_1 repasse à l'état « prêt ».
- L'Ordonnanceur élit alors le processus P_2 , qui demande la ressource R_2 . Celle-ci est libre, donc il l'obtient.
Le processus P_2 repasse à l'état « prêt ».
- Le processus P_1 redevient élu, il demande la ressource R_2 sans libérer la ressource R_1 . La ressource R_2 n'a pas été libérée par P_2 , donc il ne l'obtient pas. Le processus P_1 repasse à l'état « prêt ».
- Le processus P_2 redevient élu, il demande la ressource R_1 sans libérer la ressource R_2 . La ressource R_1 n'a pas été libérée par P_1 , donc il ne l'obtient pas. Le processus P_2 repasse à l'état « prêt ».

C'est la situation d'interblocage.



Exemple 1 : On prend deux processus P_1 et P_2 qui pourront exécuter leurs instructions à tour de rôle (P_1, P_2, \dots)

La ressource A contient « 4 » initialement et la ressource B contient « 2 ».

Actions du processus 1

1. Prendre le contrôle exclusif de la ressource A
2. Prendre le contrôle exclusif de la ressource B
3. Rajouter « 2 » à la fin de la ressource A
4. Rajouter le contenu de la ressource A au contenu de la ressource B

Actions du processus 2

1. Prendre le contrôle exclusif de la ressource A
2. Prendre le contrôle exclusif de la ressource B
3. Rajouter « 8 » au contenu de la ressource B
4. Rajouter le contenu de la ressource B au contenu de la ressource A

On commence par exemple avec la première instruction du processus 1.

Expliquer si cette situation mènera à un interblocage, ou non.

Donner l'ordre des instructions qui pourront s'effectuer jusqu'au blocage ou jusqu'à la fin du déroulement des processus.

Lorsqu'un processus attend indéfiniment une ressource on dit qu'il est dans une situation de famine.

Quatre conditions sont nécessaires à l'interblocage:

- Exclusion mutuelle : Les ressources ne sont pas partageables, un seul processus à la fois peut utiliser la ressource.
- Possession et attente : Les processus qui détiennent des ressources peuvent en demander d'autres.
- Sans préemption : Les ressources ne sont pas préemptibles c'est-à-dire que les libérations sont faites volontairement par les processus. On ne peut pas forcer un processus à rendre une ressource.
- Attente circulaire : Il doit exister un ensemble de processus P tel que P_i attend une ressource possédée par P_{i+1} .

Un **thread** (fil d'exécution, tâche, processus léger, fil d'instruction, processus allégé, exétron, unité d'exécution ou unité de traitement) est similaire à un processus car tous deux représentent l'exécution d'un ensemble d'instructions du langage machine d'un processeur. Du point de vue de l'utilisateur, ces exécutions semblent se dérouler en parallèle. Toutefois, là où chaque processus possède sa propre mémoire virtuelle, les threads d'un même processus se partagent sa mémoire virtuelle. Par contre, tous les threads possèdent leur propre pile d'exécution.

3.2.2.5. LES PROCESSUS SOUS WINDOWS

Microsoft Windows est architecturé en services (processus) fonctionnant en arrière-plan. Il est possible d'afficher facilement la liste des processus en cours dans le gestionnaire des tâches en appuyant simultanément sur « **CTRL** » + « **SHIFT** » + « **ESC** », ou sur « **CTRL** » + « **ALT** » + « **SUPPR** » puis « **Gestionnaire de tâche** » ou en exécutant directement « **taskmgr.exe** » dans la barre de recherche ou dans l'invité de commande.

Si vous lancez le gestionnaire de tâche pour la première fois, vous serez peut-être amené à cliquer sur « Plus de détails » en bas à Gauche de la fenêtre.

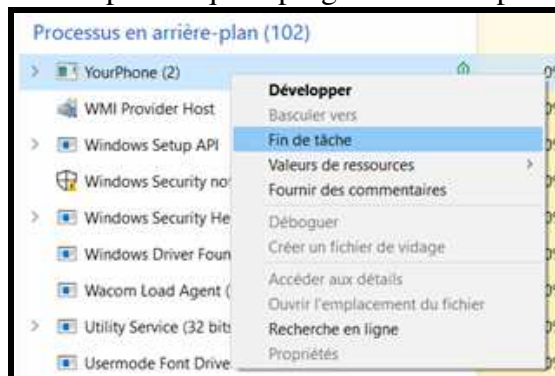
Vous verrez alors une fenêtre avec sept onglets : « Processus », « Performance », « Historique des applications », « Démarrage », « Utilisateurs », « Détails » et « Services ».

Dans l'onglet « **Processus** », vous verrez tous ceux s'exécutant sur votre machine, ainsi que les ressources qu'ils consomment. On distingue trois catégories :

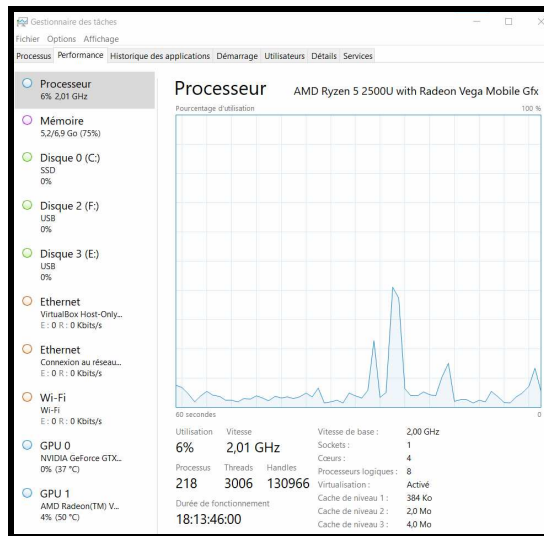
- « Applications »
- « Processus en arrière-plan »
- « Processus Windows »

Nom	Statut	6% Processeur	75% Mémoire	0% Disque	0% Réseau	13% Processe...	Moteur de processeur graphique	Consommatio...	Tendance de c...
Applications (5)									
> Microsoft Office Word (32 bits)		0%	24,6 Mo	0 Mo/s	0 Mbits/s	0%		Très faible	Très faible
> Google Chrome (16)		0%	557,5 Mo	0 Mo/s	0,1 Mbits/s	0%	GPU 1 - 3D	Très faible	Très faible
> Gestionnaire des tâches		0,5%	27,5 Mo	0 Mo/s	0 Mbits/s	0%		Très faible	Très faible
> Explorateur Windows		0,3%	41,6 Mo	0,1 Mo/s	0 Mbits/s	0%		Très faible	Très faible
> Capture d'écran et croquis (3)		0,5%	24,4 Mo	0,1 Mo/s	0 Mbits/s	1,1%	GPU 1 - 3D	Très faible	Très faible
Processus en arrière-plan (107)									
> YourPhone (2)		0%	1,2 Mo	0 Mo/s	0 Mbits/s	0%		Très faible	Très faible

Avec un clic droit sur le nom du processus et un clic sur « Fin de tâche » on peut « tuer » un processus, par exemple lorsqu'un programme ne répond plus :



Dans l'onglet « Performance », nous pouvons voir graphiquement les ressources utilisées :



Dans l'onglet « Historique des applications », nous pouvons voir les ressources utilisées par application :

The screenshot shows the Windows Task Manager History tab. It displays a table of resource usage for various applications over the last 60 seconds. The table has columns for Nom, Temps processeur, Réseau, Réseau facturé à..., and Mises à jour de... The data is as follows:

Nom	Temps processeur	Réseau	Réseau facturé à...	Mises à jour de...
Alarmes et horloge	00:00	0 Mo	0 Mo	0 Mo
AMD Radeon Software	00:00	0 Mo	0 Mo	0 Mo
Astuces	00:00	0 Mo	0 Mo	0 Mo
AudioWizard	00:00	0 Mo	0 Mo	0 Mo
Calculatrice	00:00	0 Mo	0 Mo	0 Mo
Caméra	00:00	0 Mo	0 Mo	0 Mo
Canon Inkjet Print Utility	00:00	0 Mo	0 Mo	0 Mo
Capture d'écran et croquis	00:16	0 Mo	0 Mo	0 Mo
Cartes	00:00	0 Mo	0 Mo	0 Mo
Compagnon de la console	00:00	0 Mo	0 Mo	0 Mo
Cortana	00:00	0 Mo	0 Mo	0 Mo
Courrier et calendrier	00:53	0 Mo	0 Mo	0 Mo

Dans l'onglet « Utilisateur », nous pouvons voir tous les utilisateurs connectés sur la machine et les processus actifs. Avec un clic droit sur le nom de l'utilisateur on peut le déconnecter. Avec un clic droit sur le nom du processus et un clic sur « Fin de tâche » on peut « tuer » un processus :

The screenshot shows the Windows Task Manager Users tab. It displays a table of active users and their processes. The table has columns for Utilisateur, Statut, Processeur, Mémoire, Disque, Réseau, and Moteur de pro... The data is as follows:

Utilisateur	Statut	Processeur	Mémoire	Disque	Réseau	Moteur de pro...
utilisateur	5%	75%	0%	0%	1%	
		2.9%	1 128.8 Mo	0.7 Mo/s	0 Mbits/s	0% GPU 1 - 3D

A context menu is open over the 'utilisateur' row, showing options: Développer, Déconnecter, and Gérer les comptes d'utilisateurs.

Dans l'onglet « Détails », nous pouvons voir le PID (Process Identifier : identifiant de processus) des processus et leurs statuts :

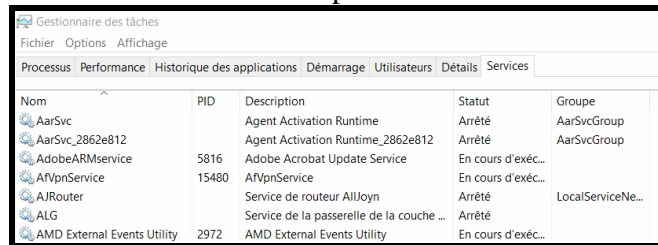
The screenshot shows the Windows Task Manager Details tab. It displays a table of process details. The table has columns for Nom, PID, Statut, Nom d'utili..., Processeur, Mémoire (plage de travail privée active), and Virtualisation du contrôle de compte d'u... The data is as follows:

Nom	PID	Statut	Nom d'utili...	Processeur	Mémoire (plage de travail privée active)	Virtualisation du contrôle de compte d'u...
AdminService.exe	5668	En cours d'exécution	Système	00	648 Ko	Non autorisé
ApplicationFrameHost.exe	17344	En cours d'exécution	utilisateur	00	9 760 Ko	Désactivé
armsvc.exe	5816	En cours d'exécution	Système	00	36 Ko	Non autorisé

Exécutez la calculatrice Windows, relever son PID et son statut.

Réduisez la calculatrice dans la barre des tâches. Que constatez-vous ?

Dans l'onglet « Services », nous pouvons voir tous les services et leurs statuts (un service est un processus qui est démarré par le système d'exploitation et qui s'exécute dans une session distincte. Un service est un processus sans interface utilisateur :



Nom	PID	Description	Statut	Groupe
AarSvc		Agent Activation Runtime	Arrêté	AarSvcGroup
AarSvc_2862e812		Agent Activation Runtime_2862e812	Arrêté	AarSvcGroup
AdobeARMSvc	5816	Adobe Acrobat Update Service	En cours d'exéc...	
AFVpnService	15480	AFVpnService	En cours d'exéc...	
AJRouter		Service de routeur AllJoyn	Arrêté	LocalServiceNe...
ALG		Service de la passerelle de la couche ...	Arrêté	
AMD External Events Utility	2972	AMD External Events Utility	En cours d'exéc...	

REMARQUES :

- Parmi ces processus un grand nombre sont des processus système faisant partie intégrante de Windows et certains correspondent à des applications tierces. Ainsi lorsque le système d'exploitation semble « ralentir » il peut être intéressant de déterminer quel est le processus consommant le plus de ressources.
- D'autre part la présence de vers, virus, chevaux de Troie, spywares, et AdWares sur le système est généralement trahie par la présence de processus suspects, c'est la raison pour laquelle ils prennent souvent un nom proche d'un processus système réel afin de passer inaperçu (par exemple system32.exe au lieu de system32.dll, isass.exe au lieu de lsass.exe).
- Cependant Microsoft fournit des outils plus complets pour l'étude des processus Windows comme Process Explorer et Process Monitor
- Pour télécharger « Process Explorer » : <https://docs.microsoft.com/fr-fr/sysinternals/downloads/process-explorer>
- Pour télécharger « Process Monitor » : <https://docs.microsoft.com/en-us/sysinternals/downloads/procmon>
- « Process Explorer » est un gestionnaire de tâches évolué qui permet de faire beaucoup de choses. La fonction première est d'afficher et lister les processus en cours d'exécution. Mais on peut aussi les terminer ou les suspendre
- « Process Explorer » va encore plus loin car il donne accès aux handles (un handle est un ID identifiant unique d'un objet Windows), DLL (bibliothèque qui contient du code et des données pouvant être utilisés simultanément par plusieurs processus) chargées / ouvertes par chaque processus. Enfin l'outil peut aussi aider à visualiser l'utilisation CPU, mémoire, GPU et disque globale et ce par application.

3.2.2.6.LES PROCESSUS SOUS LINUX

3.2.2.6.1. SYSTEME LINUX (rappels de classe de première)

Linux s'est développée de manière très rapide et a donné lieu à de nombreuses distributions partageant le même noyau.



On peut citer par exemple **UBUNTU** et **DEBIAN** parmi les plus connus. Le noyau **LINUX** contient des millions de lignes de code, pour la plupart écrite en langage C. Le système **ANDROÏD** est également basé sur le noyau **LINUX**.



3.2.2.6.2. **BASH** (rappels de classe de première)

Quelle que soit la distribution Linux utilisée, on trouve une application « terminal » ou console qui est l'interpréteur en ligne de commande de type script : **BASH** (**B**ourne **A**gain **S**hell se prononce presque comme born again (nouveau né ou réincarné), to bash (frapper violemment), Stephen Bourne créateur du BASH chez AT&T Bell version 7 d'Unix). C'est le plus courant sous Linux et aussi sous Mac OS X. Le programme avec lequel vous interagissez pour exécuter les commandes s'appelle le **Shell**.

Afin de tester les commandes BASH sans ordinateur sous système d'exploitation Linux, il est possible :

* de l'activer sous Windows 10 :

- Activer le mode développeur de Windows 10
- Installer le sous-système Windows pour Linux
- Choisir une distribution Linux pour Windows







* ou d'activer un simulateur en ligne JSLinux : <https://jslinux.org/>

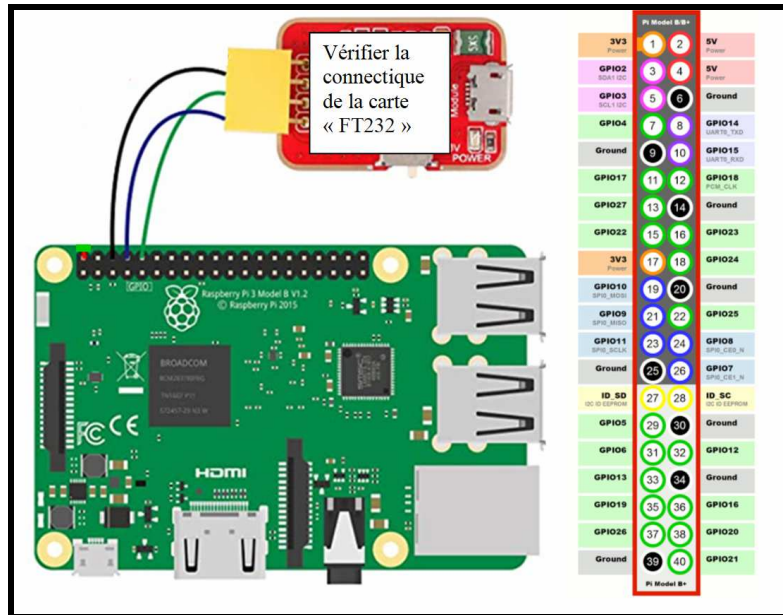
* ou d'obtenir une machine virtuelle Linux sous système d'exploitation windows. Vous trouverez la vidéo pour installer Virtualbox et Debian : <https://www.youtube.com/watch?v=m46IWiNNHbY&t=525s>

* ou d'utiliser une carte Raspberry Pi 2, 3, 4, ... :

- Créer la carte micro SD avec le logiciel « Raspberry Pi Imager » de la fondation RaspberryPi (<https://www.raspberrypi.org/software/>). Vous avez le choix entre deux versions :
 - Raspbian classique : archive de 2Go, interface graphique, nombreux logiciels pré-installés,
 - Raspbian « lite » : archive de 400Mo, mode console, minimum vital

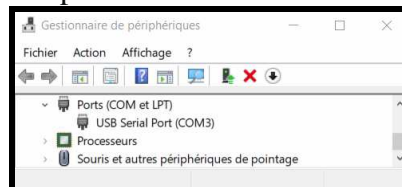
Pour notre utilisation, on prendra donc la version « lite » en choisissant l'OS « Raspberry Pi OS (other) » puis la version « lite »

- « Graver » alors la carte micro SD amorçable avec un adaptateur USB-Lecteur de carte micro SD
- A partir de la version 3 de la Raspberry Pi, vous devez activer le port série. Pour cela, ouvrez le fichier `config.txt` de la carte micro SD dans la partition `boot` et ajoutez la ligne suivante sur une nouvelle ligne en fin de script : `enable_uart=1`
-    Mettre tout **hors tension**
- Mettre en place la carte micro SD dans le lecteur de la carte Raspberry Pi
- Connecter la carte « FT232 » adaptateur port USB vers liaison série de la carte Raspberry Pi tel que :
 -    Vérifier que le cavalier ou l'interrupteur de la carte « FT232 » soit sur 3,3v
 - Faire les connexions suivantes en vous aidant du schéma et du tableau ci-dessous (les connexions TXD-RXD sont croisées). Vérifier la connectique de la carte « FT232 »

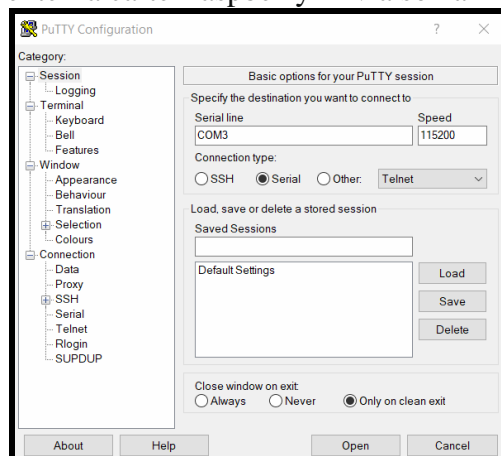



Carte « FT232 »	Carte Raspberry Pi
GND	GND broche 6
RXD	TXD broche 8
TXD	RXD broche 10

- Connecter la carte « FT232 » adaptateur port USB vers liaison série à l'ordinateur via le câble USB – mini USB.
- Installer puis ouvrir le logiciel « Putty » : <https://www.putty.org/>
- A l'aide du gestionnaire de périphériques de Windows, recherchez le numéro du port (COM) sur lequel est connecté la carte « FT232 » adaptateur port USB vers liaison série



- Ouvrir le logiciel « Putty », sélectionner « serial », mettre la vitesse de transmission « speed » à 115200, sélectionner le port (COM) sur lequel est connecté la carte « FT232 » adaptateur port USB vers liaison série (dans cet exemple : COM3), cliquer sur « open » puis alimenter la carte Raspberry Pi via son alimentation secteur



- La carte Raspberry Pi démarre et vous demande :
 - Le login : pi
 - Le mot de passe : raspberry
 -  **Ne jamais mettre hors tension la carte Raspberry Pi sous peine de détruire la carte micro SD** : Pour mettre hors tension vous devez saisir la ligne suivante afin d'éteindre la carte Raspberry Pi :
`sudo shutdown -h now`
Puis vous pouvez débrancher l'alimentation secteur de la Raspberry Pi.

`sudo` (abréviation de **substitute user do**) est une commande permettant à l'administrateur système d'accorder à certains utilisateurs (ou groupes d'utilisateurs) la possibilité de lancer une commande en tant qu'administrateur. Il permet de prendre les droits du « superutilisateur » qui a tous les droits : root. C'est l'administrateur de la machine, le seul à être autorisé à installer des programmes ou effectuer certaines modifications sur le système.

`shutdown -h` permet d'éteindre le système

`shutdown -r` permet de redémarrer le système

`now` immédiatement

`0` immédiatement

`10` au bout de 10 minutes

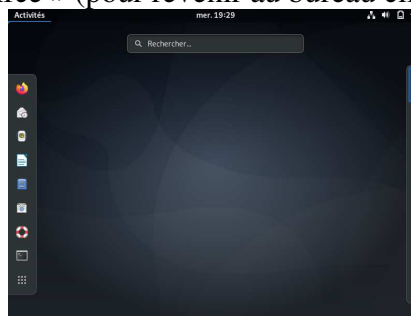
3.2.2.6.3. LES PROCESSUS SOUS LINUX

Sous linux, les deux principales commandes pour visualiser les processus sont « top » et « ps », la différence étant que l'on utilise « top » dans un terminal et « ps » dans un script où il est combiné à d'autres commandes.

À la différence de la partie précédente sous Windows, nous allons travailler en ligne de commandes.

Relisez le cours de première « les systèmes d'exploitation » sur les commandes linux.

Lancer la machine virtuelle et ouvrir un terminal avec un clic sur « Activités » et un sur « Terminal Xfce » (pour revenir au bureau cliquez sur X) :



3.2.2.6.3.1. COMMANDE TOP SOUS LINUX

Saisir la commande « top » (pour Table Of Processes), elle est l'équivalent du gestionnaire de tâches de Windows. Elle apporte donc des renseignements sur la consommation mémoire, CPU, tous les processus en cours,... Elle apporte des statistiques de consommation en temps réel, une mise à jour est effectuée toutes les 5 secondes environ :

```

top - 19:34:38 up 10 min, 1 user, load average: 0.07, 0.06, 0.04
Tasks: 146 total, 2 running, 144 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0.5 us, 0.2 sy, 0.0 ni, 98.9 id, 0.1 wa, 0.0 hi, 0.2 si, 0.0 st
MiB Mem : 3803.0 total, 1857.2 free, 585.0 used, 560.8 buff/cache
MiB Swap: 975.0 total, 975.0 free, 0.0 used, 2258.2 avail Mem

  PID USER      PR  NI   VIRT   RES   SHR  S  %CPU  %MEM     TIME+ COMMAND
 810 lf        20   0 3127752 213760 100720 S   2.6   7.0   0:34.90 gnome-s+
1667 lf        20   0 382976 37572 28644 S   0.3   1.2   0:00.19 xfce4-t+
1679 lf        20   0 11148 3532 3096 R   0.3   0.1   0:00.02 top
   1 root       20   0 169536 10184 7828 S   0.0   0.3   0:01.72 systemd
   2 root       20   0 0 0 0 S   0.0   0.0   0:00.01 kthreadd
   3 root       20  -20 0 0 0 I   0.0   0.0   0:00.00 rcu_gp
   4 root       20  -20 0 0 0 I   0.0   0.0   0:00.00 rcu_par+
   6 root       20  -20 0 0 0 I   0.0   0.0   0:00.00 kworker+
   7 root       20   0 0 0 0 I   0.0   0.0   0:00.03 kworker+
   8 root       20  -20 0 0 0 I   0.0   0.0   0:00.00 mm_perc+
   9 root       20   0 0 0 0 S   0.0   0.0   0:00.03 ksoftir+
  10 root       20   0 0 0 0 I   0.0   0.0   0:00.11 rcu_sch+
  11 root       20   0 0 0 0 I   0.0   0.0   0:00.00 rcu_bh
  12 root       20   0 0 0 0 S   0.0   0.0   0:00.00 migrati+
  13 root       20   0 0 0 0 R   0.0   0.0   0:00.27 kworker+
  14 root       20   0 0 0 0 S   0.0   0.0   0:00.00 cpuhp/0
  15 root       20   0 0 0 0 S   0.0   0.0   0:00.00 cpuhp/1
  16 root       20   0 0 0 0 S   0.0   0.0   0:00.33 migrati+
  17 root       20   0 0 0 0 S   0.0   0.0   0:00.01 ksoftir+

```

Voici quelques commandes que vous pouvez exécuter directement au clavier lorsque la commande « top » est exécutée:

- « H » ou « ? » permet d'afficher une fenêtre d'aide avec toutes les commandes et informations utiles.
- « Espace » permet de mettre à jour la liste des processus.
- « F » permet d'ajouter ou de supprimer des champs.
- « Q » permet de quitter l'application « top » ou une fenêtre rattachée à « top ». Par exemple, après avoir utilisé la fonction « H »
- « M » permet d'afficher des informations sur la mémoire.
- ...

Ajoutez la colonne qui donne le « PPID » (Parent Process id).

Remplacez la colonne « Time+ » par la colonne « Time ».

Lancez Firefox puis relevez son « PID » et le nom de « COMMAND » qui lui est associé.

3.2.2.6.3.2. COMMANDE PIDOF SOUS LINUX

« pidof » cherche les PID (Process ID) des processus portant l'un des noms de programme indiqués.

Il suffit de saisir « pidof » suivi du nom de COMMAND du processus.

Lancez Firefox puis relevez ses « PID ».

3.2.2.6.3.3. COMMANDE KILL SOUS LINUX

Cette commande permet d'arrêter un processus.

Il suffit de saisir « kill » suivi du « PID » du processus que l'on veut arrêter.

Lancez Firefox puis relevez ses « PID » avec la commande « pidof ».

Exécutez la commande « kill » sur chaque « PID ». A chaque fois, avec la commande « pidof », vérifiez l'action de la commande « kill » et le fonctionnement de firefox.

Faites des recherches sur l'utilisation de « pkill ». Lancez Firefox puis exécutez la commande « pkill ». Vérifiez l'arrêt du fonctionnement de firefox.

3.2.2.6.3.4. COMMANDE PS SOUS LINUX

Jusqu'à présent nous avons utilisé la commande « top » pour visualiser l'ensemble des processus en temps réel.

La commande « ps » (Processes Status) permet aussi d'afficher l'ensemble des processus en cours d'exécution sous forme arborescente à un temps donné (pas de mise à jour en temps réel).

« ps aux » :

- a : montre les processus de tous les utilisateurs.
- u : montre le propriétaire des processus.
- x : montre les processus non rattachés à un terminal.

S'il y a trop d'informations, il est possible de filtrer avec la commande « grep ». Si je recherche le processus « nsi » je tape la commande « ps aux | nsi »

Exécutez la commande « ps aux » en ligne de commande.

Comparez avec le résultat obtenu avec la commande « top ».

Exécutez la commande « ps -ef » en ligne de commande.

Quelle est la différence entre la commande « ps -ef » et « ps aux » ?

3.2.3. LES PROCESSUS A L'EPREUVE ECRITE

3.2.3.1. SUJET 1, EXERCICE 2

Cet exercice porte sur la gestion des processus par les systèmes d'exploitation et sur les opérateurs booléens.

Partie A

Cette partie est un questionnaire à choix multiples (QCM).

Pour chacune des questions, une seule des quatre réponses est exacte. Le candidat indiquera sur sa copie le numéro de la question et la lettre correspondant à la réponse exacte.

Aucune justification n'est demandée. Une réponse fausse ou une absence de réponse n'enlève aucun point.

1. Parmi les commandes ci-dessous, laquelle permet d'afficher les processus en cours d'exécution ?

- a. dir
- b. ps
- c. man
- d. ls

2. Quelle abréviation désigne l'identifiant d'un processus dans un système d'exploitation de type UNIX ?

- a. PIX
- b. SIG
- c. PID
- d. SID

3. Comment s'appelle la gestion du partage du processeur entre différents processus ?

- a. L'interblocage
- b. L'ordonnancement
- c. La planification
- d. La priorisation

4. Quelle commande permet d'interrompre un processus dans un système d'exploitation de type UNIX ?

- a. stop
- b. interrupt
- c. end
- d. kill

Partie B

1. Un processeur choisit à chaque cycle d'exécution le processus qui doit être exécuté. Le tableau ci-dessous donne pour trois processus P1, P2, P3 :

- la durée d'exécution (en nombre de cycles),
- l'instant d'arrivée sur le processeur (exprimé en nombre de cycles à partir de 0),
- le numéro de priorité.

Le numéro de priorité est d'autant plus petit que la priorité est grande. On suppose qu'à chaque instant, c'est le processus qui a le plus petit numéro de priorité qui est exécuté, ce qui peut provoquer la suspension d'un autre processus, lequel reprendra lorsqu'il sera le plus prioritaire.

Processus	Durée d'exécution	Instant d'arrivée	Numéro de priorité
P1	3	3	1
P2	3	2	2
P3	4	0	3

Reproduire le tableau ci-dessous sur la copie et indiquer dans chacune des cases le processus exécuté à chaque cycle

P3										
0	1	2	3	4	5	6	7	8	9	10

2. On suppose maintenant que les trois processus précédents s'exécutent et utilisent une ou plusieurs ressources parmi R1, R2 et R3. Parmi les scénarios suivants, lequel provoque un interblocage ? Justifier.

Scénario 1
P1 acquiert R1
P2 acquiert R2
P3 attend R1
P2 libère R2
P2 attend R1
P1 libère R1

Scénario 2
P1 acquiert R1
P2 acquiert R3
P3 acquiert R2
P1 attend R2
P2 libère R3
P3 attend R1

Scénario 3
P1 acquiert R1
P2 acquiert R2
P3 attend R2
P1 attend R2
P2 libère R2
P3 acquiert R2

3.2.3.2.SUJET 2 , EXERCICE 4

Cet exercice porte sur les systèmes d'exploitation : gestion des processus et des ressources.

Les parties A et B peuvent être traitées indépendamment.

Partie A :

Dans un bureau d'architectes, on dispose de certaines ressources qui ne peuvent être utilisées simultanément par plus d'un processus, comme l'imprimante, la table traçante, le modem. Chaque programme, lorsqu'il s'exécute, demande l'allocation des ressources qui lui sont nécessaires. Lorsqu'il a fini de s'exécuter, il libère ses ressources.

Programme 1	Programme 2	Programme 3
Demander (table traçante)	demander (modem)	demander (imprimante)
demander (modem)	demander (imprimante)	demander (table traçante)
exécution	exécution	exécution
libérer (modem)	libérer (imprimante)	libérer (table traçante)
libérer (table traçante)	libérer (modem)	libérer (imprimante)

On appelle p1, p2 et p3 les processus associés respectivement aux programmes 1, 2 et 3.

1. Les processus s'exécutent de manière concurrente. Justifier qu'une situation d'interblocage peut se produire.
2. Modifier l'ordre des instructions du programme 3 pour qu'une telle situation ne puisse pas se produire. Aucune justification n'est attendue.
3. Supposons que le processus p1 demande la table traçante alors qu'elle est en cours d'utilisation par le processus p3. Parmi les états suivants, quel sera l'état du processus p1 tant que la table traçante n'est pas disponible :
a) élu b) bloqué c) prêt d) terminé

Partie B :

Avec une ligne de commande dans un terminal sous Linux, on obtient l'affichage suivant :

UID	PID	PPID	C	STIME	TTY	TIME	CMD
...							
pi	6211	831	8	09:07	?	00:01:16	/usr/lib/chromium-browser/chromium-browser-v7 --disable-quit --enable-tcp-fast-open --p
pi	6252	6211	0	09:07	?	00:00:00	/usr/lib/chromium-browser/chromium-browser-v7 --type=zygote --ppapi-flash-path=/usr/lib
pi	6254	6252	0	09:07	?	00:00:00	/usr/lib/chromium-browser/chromium-browser-v7 --type=zygote --ppapi-flash-path=/usr/lib
pi	6294	6211	4	09:07	?	00:00:40	/usr/lib/chromium-browser/chromium-browser-v7 --type=gpu-process --field-trial-handle=1
pi	6300	6211	1	09:07	?	00:00:16	/usr/lib/chromium-browser/chromium-browser-v7 --type=utility --field-trial-handle=10756
pi	6467	6254	1	09:07	?	00:00:11	/usr/lib/chromium-browser/chromium-browser-v7 --type=renderer --field-trial-handle=1075
pi	11267	6254	2	09:12	?	00:00:15	/usr/lib/chromium-browser/chromium-browser-v7 --type=renderer --field-trial-handle=1075
pi	12035	836	0	09:13	?	00:00:00	/usr/lib/Libreoffice/program/oosplash --writer file:///home/pi/Desktop/mon_fichier.odt
pi	12073	12035	2	09:13	?	00:00:15	/usr/lib/Libreoffice/program/soffice.bin --writer file:///home/pi/Desktop/mon_fichier.c
pi	12253	831	1	09:13	?	00:00:07	/usr/bin/python3 /usr/bin/sense_emu_gui
pi	20010	6211	1	09:21	?	00:00:00	/usr/lib/chromium-browser/chromium-browser-v7 --type=utility --field-trial-handle=10756
pi	20029	6254	56	09:21	?	00:00:28	/usr/lib/chromium-browser/chromium-browser-v7 --type=renderer --field-trial-handle=1075
pi	20339	6254	4	09:21	?	00:00:01	/usr/lib/chromium-browser/chromium-browser-v7 --type=renderer --field-trial-handle=1075
pi	20343	6254	2	09:21	?	00:00:00	/usr/lib/chromium-browser/chromium-browser-v7 --type=renderer --field-trial-handle=1075
pi	20464	6211	17	09:22	?	00:00:00	/proc/self/exe --type=utility --field-trial-handle=1075863133478894917,6306120996223181
pi	20488	6254	14	09:22	?	00:00:00	/usr/lib/chromium-browser/chromium-browser-v7 --type=renderer --field-trial-handle=1075
pi	20519	676	0	09:22	pts/0	00:00:00	ps -ef

La documentation Linux donne la signification des différents champs :

- UID : identifiant utilisateur effectif ;
- PID : identifiant de processus ;
- PPID : PID du processus parent ;
- C : partie entière du pourcentage d'utilisation du processeur par rapport au temps de vie des processus ;
- STIME : l'heure de lancement du processus ;
- TTY : terminal de contrôle
- TIME : temps d'exécution
- CMD : nom de la commande du processus

1. Parmi les quatre commandes suivantes, laquelle a permis cet affichage ?
 - a) ls -l
 - b) ps -ef
 - c) cd ..
 - d) chmod 741 processus.txt
2. Quel est l'identifiant du processus parent à l'origine de tous les processus concernant le navigateur Web (chromium-browser) ?
3. Quel est l'identifiant du processus dont le temps d'exécution est le plus long ?

3.2.3.3.SUJET 3 , EXERCICE 4

Cet exercice traite principalement du thème « architectures matérielles, systèmes d'exploitation et réseaux ». Cet exercice mobilise des connaissances sur le routage et sur l'évolution des architectures des machines.

Partie B : Système d'exploitation

Un système d'exploitation doit assurer la gestion des processus et des ressources.

1. Dans ce contexte, expliquer et illustrer par un exemple ce qu'est une situation d'interblocage (deadlock).
2. Citer des mécanismes permettant d'éviter ces situations.

3.2.3.4.SUJET 4 , EXERCICE 3

Cet exercice porte sur la gestion des processus par un système d'exploitation et les protocoles de routage.

Les parties A et B sont indépendantes.

Partie A : Processus

La commande UNIX `ps` présente un cliché instantané des processus en cours d'exécution.

Avec l'option `-eo pid,ppid,stat,command`, cette commande affiche dans l'ordre l'identifiant du processus PID (process identifier), le PPID (parent process identifier), l'état STAT et le nom de la commande à l'origine du processus. Les valeurs du champ STAT indique l'état des processus :

R : processus en cours d'exécution

S : processus endormi

Sur un ordinateur, on exécute la commande `ps -eo pid,ppid,stat,command` et on obtient un affichage dont on donne ci-dessous un extrait.

```
$ ps -eo pid,ppid,stat,command
PID      PPID      STAT      COMMAND
1         0         Ss        /sbin/init
.....
1912     1908     Ss        Bash
2014     1912     Ss        Bash
1920     1747     Sl        Gedit
2013     1912     Ss        Bash
2091     1593     Sl        /usr/lib/firefox/firefox
5437     1912     Sl        python programme1.py
5440     2013     R         python programme2.py
5450     1912     R+        ps -eo pid,ppid,stat,command
```

À l'aide de cet affichage, répondre aux questions ci-dessous.

1. Quel est le nom de la première commande exécutée par le système d'exploitation lors du démarrage ?
2. Quels sont les identifiants des processus actifs sur cet ordinateur au moment de l'appel de la commande `ps` ? Justifier la réponse.
3. Depuis quelle application a-t-on exécuté la commande `ps` ? Donner les autres commandes qui ont été exécutées à partir de cette application.
4. Expliquer l'ordre dans lequel les deux commandes `python programme1.py` et `python programme2.py` ont été exécutées.
5. Peut-on prédire que l'une des deux commandes `python programme1.py` et `python programme2.py` finira avant l'autre ?

3.2.3.5. SUJET 5, EXERCICE 2

Cet exercice porte sur la gestion des processus et des ressources par un système d'exploitation.

1. Les états possibles d'un processus sont : prêt, élu, terminé et bloqué.

a. Expliquer à quoi correspond l'état élu.

b. Proposer un schéma illustrant les passages entre les différents états.

2. On suppose que quatre processus C_1 , C_2 , C_3 et C_4 sont créés sur un ordinateur, et qu'aucun autre processus n'est lancé sur celui-ci, ni préalablement ni pendant l'exécution des quatre processus.

L'ordonnanceur, pour exécuter les différents processus prêts, les place dans une structure de données de type file. Un processus prêt est enfilé et un processus élu est défilé.

a. Parmi les propositions suivantes, recopier celle qui décrit le fonctionnement des entrées/sorties dans une file :

- i. Premier entré, dernier sorti
- ii. Premier entré, premier sorti
- iii. Dernier entré, premier sorti

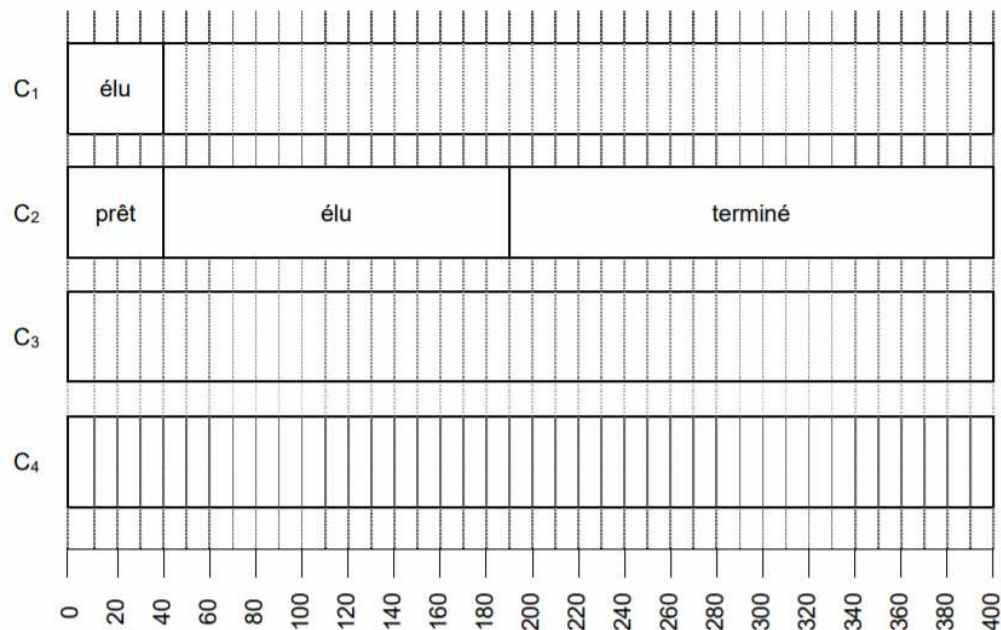
b. On suppose que les quatre processus arrivent dans la file et y sont placés dans l'ordre C₁, C₂, C₃ et C₄.

- Les temps d'exécution totaux de C₁, C₂, C₃ et C₄ sont respectivement 100 ms, 150 ms, 80 ms et 60 ms.

- Après 40 ms d'exécution, le processus C₁ demande une opération d'écriture disque, opération qui dure 200 ms. Pendant cette opération d'écriture, le processus C₁ passe à l'état bloqué.

- Après 20 ms d'exécution, le processus C₃ demande une opération d'écriture disque, opération qui dure 10 ms. Pendant cette opération d'écriture, le processus C₃ passe à l'état bloqué.

Sur la frise chronologique ci-dessous, les états du processus C₂ sont donnés. Compléter la frise avec les états des processus C₁, C₃ et C₄.



3. On trouvera ci-dessous deux programmes rédigés en pseudo-code

Verrouiller un fichier signifie que le programme demande un accès exclusif au fichier et l'obtient si le fichier est disponible.

Programme 1

```
Verrouiller fichier_1
Calculs sur fichier_1
Verrouiller fichier_2
Calculs sur fichier_1
Calculs sur fichier_2
Calculs sur fichier_1
Déverrouiller fichier_2
Déverrouiller fichier_1
```

Programme 2

```
Verrouiller fichier_2
Verrouiller fichier_1
Calculs sur fichier_1
Calculs sur fichier_2
Déverrouiller fichier_1
Déverrouiller fichier_2
```

a. En supposant que les processus correspondant à ces programmes s'exécutent simultanément (exécution concurrente), expliquer le problème qui peut être rencontré.

b. Proposer une modification du programme 2 permettant d'éviter ce problème.

4. PROTOCOLES DE ROUTAGE

4.1. NOTION D'ADRESSAGE (rappels de classe de première)

Deux types d'adresse :

* L'**adresse IP (Internet Protocol)** est un numéro d'identification qui est attribué de façon permanente ou provisoire à chaque périphérique relié à un réseau informatique qui utilise l'Internet Protocol. L'adresse IP est à la base du système d'acheminement (le routage) des paquets de données sur Internet.

Il existe deux types d'adresse IP :

- IPv4 : adresse IP codée sur 32 bits en binaire découpée 4 groupes de 8 bits.
- IPv6 : adresse IP codée sur 128 bits en hexadécimal découpée en 8 groupes de 16 bits. Elle tend à remplacer l'IPv4 qui ne propose plus assez d'adresses.

* L'**adresse MAC (Media Access Control)** est un identifiant physique stocké dans une interface réseau. Elle est unique au monde. Elle constitue la couche inférieure de la couche de « liaison », c'est-à-dire la couche 2 du modèle OSI (Open Systems Interconnection : interconnexion des systèmes ouverts, norme de 1984 pour la conception des réseaux et la fabrication des équipements). Elle est constituée de six octets, il existe donc potentiellement 2^{48} (environ 281 000 milliards) d'adresses MAC possibles.

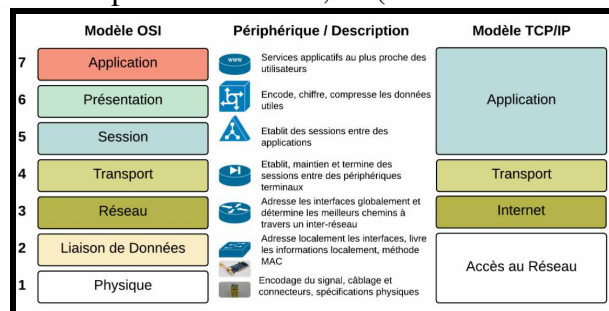
Ouvrir une console (invité de commandes : cmd) et saisir la commande « ipconfig/all » pour obtenir les adresses IP et MAC ou « getmac /v ».

Comparer avec l'IP obtenue avec celle donnée sur « monip.com »

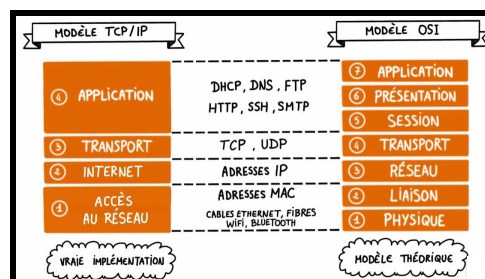
Comparer avec l'IP obtenue avec celle donnée sur « ping.eu »

4.1.1. MODELES OSI ET TCP/IP (rappels de classe de première)

OSI : En 1984, l'**I.S.O.** (International Standardization Organisation) publie la norme 7498. C'est le modèle **OSI** (Open Systems Interconnection : Interconnexion des systèmes ouverts). C'est un modèle théorique, le modèle pratique utilisé est TCP/IP. TCP/IP est protocole de transport, TCP (Transmission Control Protocol : protocole de contrôle de transmissions) qu'on utilise « par-dessus » un protocole réseau, IP (Internet Protocol : protocole Internet).



Visualisez la vidéo (grâce aux paramètres, vous pouvez si besoin ralentir la vitesse de lecture) : <https://www.youtube.com/watch?v=26jazyc7VNk>



Les machines et leurs interfaces disposent d'identifiants au niveau de chaque couche :

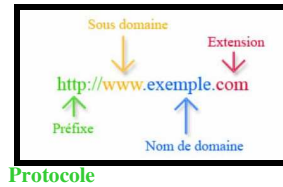
Couche	Identifiant	Exemple
Couche 4 Application	Un protocole et un nom de domaine	http:// suivi de www.cisco.com
Couche 3 Transport	Port TCP ou UDP	TCP80 comme port par défaut pour HTTP
Couche 2 Internet	Adresse IPv4 et/ou IPv6	192.168.150.252/24 ou 2001:db8::1/64
Couche 1 Accès au réseau	Adresse physique (MAC 802)	70:56:81:bf:7c:37

TCP/IP	OSI	Rôles	PDU	Protocoles	Matériel
4 Application	7 Application	Services au plus proche des utilisateurs	Données / message	HTTP, DNS, DHCP	Ordinateurs
4 Application	6 Présentation	encode, chiffre, compresse les données utiles	Données / message	HTTP, DNS, DHCP	Ordinateurs
4 Application	5 Session	établit des sessions entre des applications	Données / message	HTTP, DNS, DHCP	Ordinateurs
3 Transport	4 Transport	établit, maintient, termine des sessions entre des hôtes d'extrémité.	Segment	TCP ou UDP	Ordinateurs, routeurs NAT, pare-feux
2 Internet	3 Réseau	Identifie les hôtes et assure leur connectivité	Datagramme ou paquet	IPv4 ou IPv6	Routeurs
1 Accès réseau	2 Liaison de données	Détermine la méthode d'accès au support, organise les bits de données	Trame	Ethernet IEEE 802.3, Wi-Fi IEEE 802.11, pontage 802.1	Commutateurs, cartes d'interface réseau
1 Accès réseau	1 Physique	s'occupe du placement physique du signal	bits	Normes physiques : xDSL (WAN), 1000-BASE-TX	Câblage (UTP CAT 6) et connecteurs (RJ-45), bande fréquences (2.4 GHz, 5 GHz)

NAT (network address translation : traduction d'adresse réseau ou translation d'adresse réseau). Un routeur fait du NAT lorsqu'il fait correspondre des adresses IP à d'autres adresses IP. Il est possible de faire correspondre une seule adresse externe publique visible sur Internet à toutes les adresses d'un réseau privé, afin de pallier l'épuisement des adresses IPv4. Visualisez la vidéo (grâce aux paramètres, vous pouvez si besoin ralentir la vitesse de lecture) : <https://www.youtube.com/watch?v=jq3SLuhlyPI>

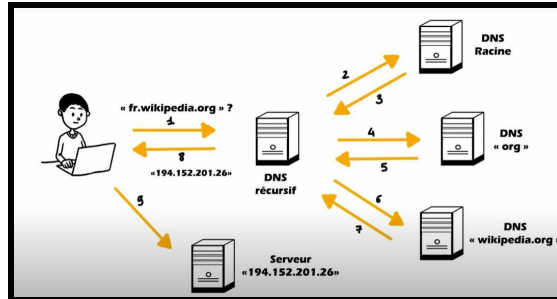
HTTP (Hypertext Transfer Protocol : protocole de transfert hypertexte) est un protocole de communication client - serveur développé pour le World Wide Web. HTTPS (S : secured, sécurisé) est la variante sécurisée par l'usage des protocoles Transport Layer Security (TLS). HTTP est un protocole de la couche application et utilise le protocole TCP comme couche de transport. Un serveur HTTP utilise alors par défaut le port 80 (443 pour HTTPS). Les clients HTTP les plus connus sont les navigateurs Web permettant à un utilisateur d'accéder à un serveur contenant les données.

DNS (Domain Name System : système de noms de domaine) permet de faciliter la recherche d'un site sur Internet. Il permet d'associer un nom compréhensible (une adresse logique : le nom de domaine) à une adresse IP (une adresse physique).



Visualisez la vidéo (grâce aux paramètres, vous pouvez si besoin ralentir la vitesse de lecture) :

<https://www.youtube.com/watch?v=qzWdzAvfBoo>



DHCP (Dynamic Host Configuration Protocol : protocole de configuration dynamique des hôtes) est un protocole réseau dont le rôle est d'assurer la configuration automatique des paramètres IP d'une machine, en lui attribuant automatiquement une adresse IP et un masque de sous - réseau. Il peut aussi configurer l'adresse de la passerelle par défaut et des serveurs de noms DNS. Visualisez la vidéo (grâce aux paramètres, vous pouvez si besoin ralentir la vitesse de lecture) :

<https://www.youtube.com/watch?v=yH9UvkeAz-I>

TCP (Transmission Control Protocol : protocole de contrôle de transmissions) Une session TCP fonctionne en trois phases :

- Etablissement de la connexion ;
- Transferts de données ;
- Fin de la connexion.

L'établissement de la connexion se fait par un « handshaking » (poignée de main) en trois temps. La rupture de connexion utilise également un « handshaking » en quatre temps. Pendant la phase d'établissement de la connexion, des paramètres comme le numéro de séquence sont initialisés afin d'assurer la transmission fiable (sans perte et dans l'ordre) des données.

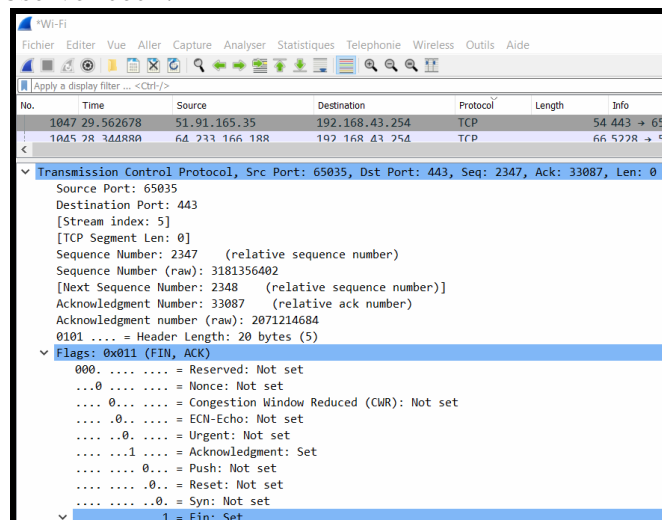
Format du message TCP :

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Port Source 2 octets																Port destination 2 octets															
Numéro de séquence																															
Numéro d'acquittement																															
Taille de l'en-tête		Réservé		ECN / NS		CWR		ECE		URG		ACK		PSH		RST		SYN		FIN		Fenêtre									
Somme de contrôle																Pointeur de données urgentes															
Options																						Remplissage									
Données																															

- Port source : numéro du port source
- Port destination : numéro du port destination
- Numéro de séquence : numéro de séquence du premier octet de ce segment
- Numéro d'acquittement : numéro de séquence du prochain octet attendu

- Taille de l'en-tête : longueur de l'en-tête en mots de 32 bits (les options font partie de l'en-tête)
- Indicateurs ou Flags (drapeaux) :
 - Réserve : réservé pour un usage futur
 - ECN/NS : Explicit Congestion Notification, signale la présence de congestion ou No Signaling, absence de signalisation
 - CWR : Congestion Window Reduced, indique qu'un paquet avec ECE a été reçu et que la congestion a été traitée
 - ECE : ECN-Echo, informe l'émetteur qu'un paquet a été reçu
 - URG : signale la présence de données **URG**entes
 - ACK : signale que le paquet est un accusé de réception (**ACK**nowledgement)
 - PSH : données à envoyer tout de suite (**PuSH**)
 - RST : rupture anormale de la connexion (**ReSeT**)
 - SYN : demande de **SYN**chronisation ou établissement de connexion
 - FIN : demande la **FIN** de la connexion
- Fenêtre : taille de fenêtre demandée, c'est-à-dire le nombre d'octets que le récepteur souhaite recevoir sans accusé de réception
- Somme de contrôle : somme de contrôle calculée sur l'ensemble de l'en-tête TCP et des données, mais aussi sur un pseudo en-tête (extrait de l'en-tête IP)
- Pointeur de données urgentes : position relative des dernières données urgentes
- Options : facultatives
- Remplissage : zéros ajoutés pour aligner les champs suivants du paquet sur 32 bits, si nécessaire
- Données : séquences d'octets transmis par l'application (par exemple : +OK POP3 server ready...)

En utilisant « Wireshark », logiciel permettant d'observer les échanges qui transitent sur un réseau, on peut observer ceci :

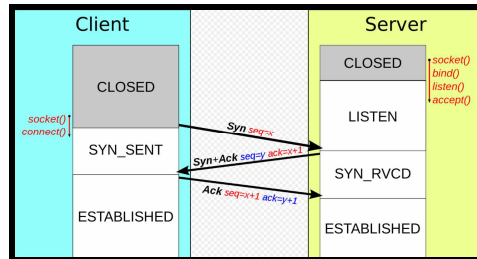


Téléchargez et ouvrez « Wireshark ». Démarrez la capture des trames avec « Wireshark » en cliquant sur « l'aileron de requin ». Surfez sur le Web, par exemple sur le site du lycée Turgot, regardez une vidéo en ligne et ouvrez « Pronote ». Arrêtez la capture des trames avec « Wireshark » en cliquant sur le carré rouge. Dans la liste des trames, trouvez les différents protocoles présents. Dans la liste des drapeaux, trouvez les différents drapeaux activés.

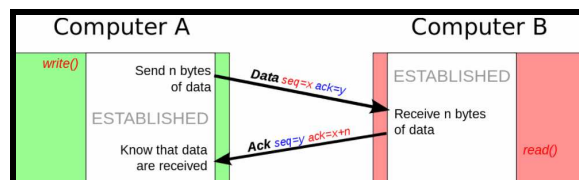
Visualisez la vidéo (grâce aux paramètres, vous pouvez si besoin ralentir la vitesse de lecture) :

<https://www.youtube.com/watch?v=04qcPyUEQE4>

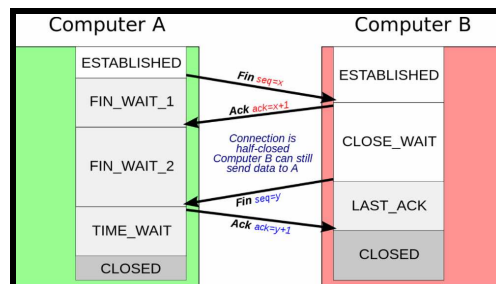
L'établissement de la connexion TCP par un « handshaking » (poignée de main) en trois temps :



Transfert de données par connexion TCP :



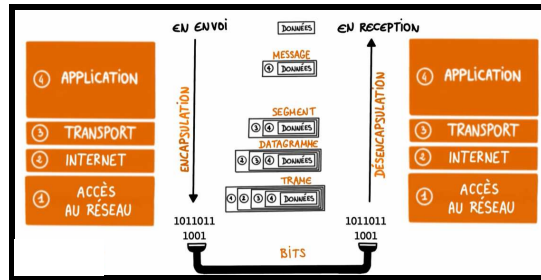
La rupture de connexion TCP par un « handshaking » en quatre temps :



UDP (User Datagram Protocol : protocole de datagrammes / paquets utilisateur) est un des principaux protocoles de télécommunication Internet. Il fait partie de la couche 4 « transport » du modèle OSI. Le rôle de ce protocole est de permettre la transmission de données de manière très simple entre deux entités, chacune étant définie par une adresse IP et un numéro de port. Ce protocole encapsule moins d'informations sur le destinataire avec les données et aucun accusé de réception n'assure la bonne réception de l'envoi. La fiabilité n'est plus garantie mais la mise en oeuvre est plus rapide (pas d'aller - retours). Aucune communication préalable n'est requise pour établir la connexion. Au contraire de TCP (qui utilise le procédé de handshaking), UDP utilise un mode de transmission sans connexion.

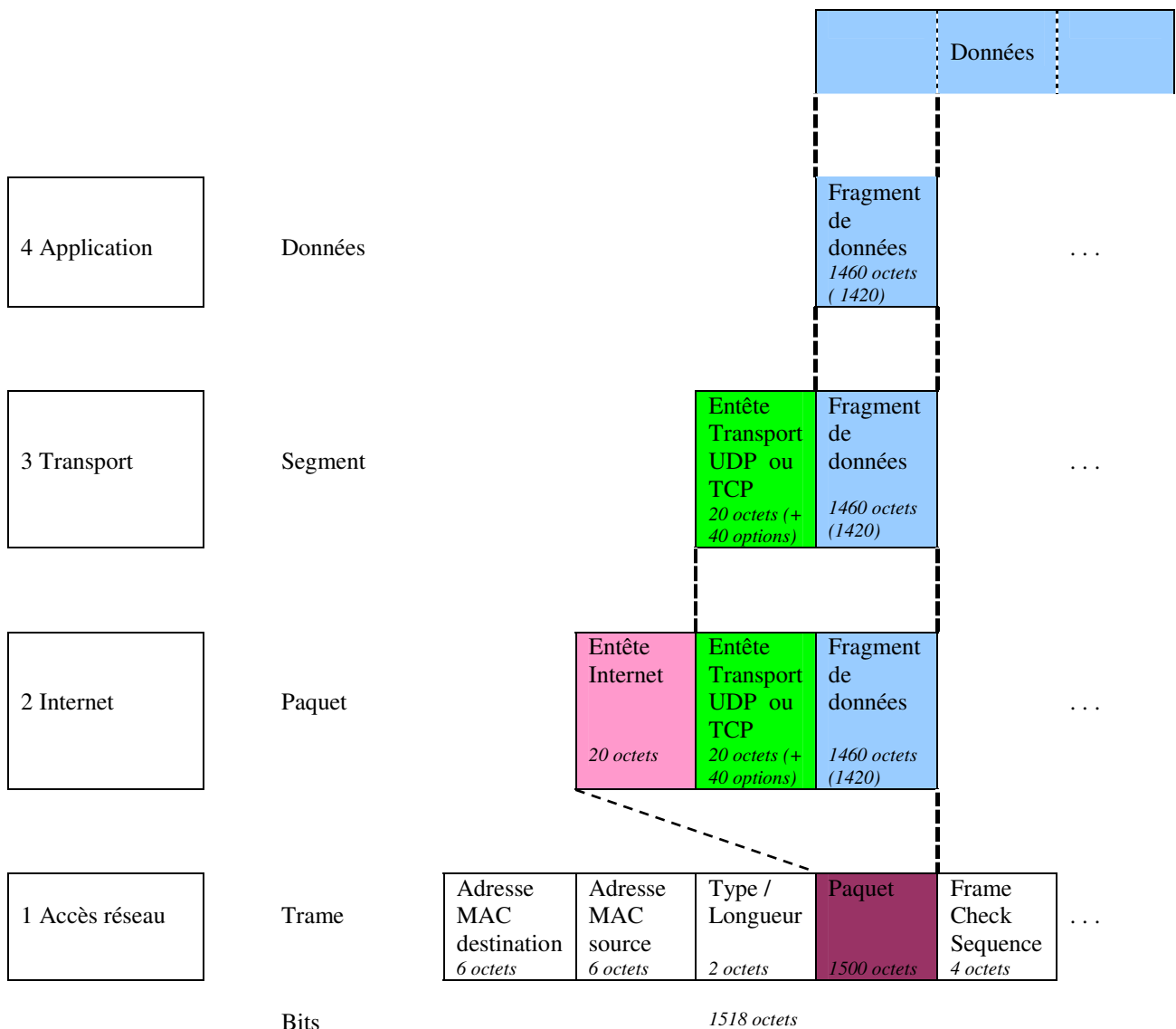
Visualisez la vidéo (grâce aux paramètres, vous pouvez si besoin ralentir la vitesse de lecture) : <https://www.it-connect.fr/les-protocoles-tcp-et-udp-pour-les-debutants/>

PDU (Protocol Data Unit : unité de données de protocole) définit l'état des données lorsqu'elles passent d'une couche du modèle TCP/IP à l'autre. Elle permet d'identifier un message en fonction de sa position. Lors de l'encapsulation (envoi) des données une entête est ajouté, elle est retirée lors de désencapsulation / décapsulation (réception).



Lors de la phase d'envoi,

- Dans la couche 4 « Application » du modèle TCP/IP, les données sont découpées, numérotées et organisées pour être envoyées.
- Dans la couche 3 « Transport » du modèle TCP/IP, des informations sont ajoutées aux données à transmettre pour permettre, entre autre, la reconstitution des données complètes par le récepteur une fois tous les paquets reçus. Cela forme le segment.
- Dans la couche 2 « Internet » du modèle TCP/IP, des informations sont ajoutées pour former le datagramme.
- Dans la couche 1 « Accès au réseau » du modèle TCP/IP, des informations sont ajoutées pour former la trame.



FTP (File Transfer Protocol : protocole de transfert de fichier) est un protocole de communication destiné au partage de fichiers sur un réseau TCP/IP.

SSH (Secure Shell : communication sécurisée) est à la fois un programme informatique et un protocole de communication sécurisé. Le protocole de connexion impose un échange de clés de chiffrement en début de connexion. Par la suite, tous les segments TCP sont authentifiés et chiffrés.

SMTP (Simple Mail Transfer Protocol : protocole simple de transfert de courrier) est un protocole de communication utilisé pour transférer le courrier électronique.

Wi-Fi (Wireless Fidelity : fidélité sans fil) est ensemble de protocoles de communication sans fil (par ondes radio) régis par les normes du groupe IEEE 802.11 (ISO/CEI 8802-11).

IEEE-802 (Institute of Electrical and Electronics Engineers : Institut d'ingénieurs en électricité et électronique) décrit une famille de normes relatives aux réseaux locaux (LAN) et métropolitains (MAN) basés sur la transmission de données numériques par des liaisons filaires ou sans fil.

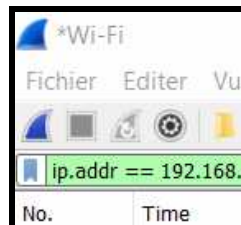
xDSL (digital subscriber line : ligne d'abonné numérique), **ADSL** (asymmetric digital subscriber line : ligne d'abonné numérique asymétrique (débit montant et descendant différents)), **SDSL** (symetric digital subscriber line : ligne d'abonné numérique symétrique (débit montant et descendant équivalents)), **ADSL2+** (ADSL améliorée) et **Re-ADSL** (reach-extended asymmetric digital subscriber line : ligne d'abonné numérique asymétrique à portée étendue (ADSL2+ améliorée))

Recherchez et notez les adresses MAC bluetooth, MAC Wi-Fi et IP de votre téléphone portable (Pour les androïdes : « paramètres », « A propos du téléphone », « Toutes les spécifications » puis « Etat »).

Recherchez et notez les adresses MAC bluetooth, MAC Wi-Fi, MAC Ethernet, ... et IP de votre ordinateur (Ouvrir une console (invité de commandes : cmd) et saisir la commande « ipconfig/all » pour obtenir les adresses IP et MAC ou « getmac /v »).

Démarrez la capture de paquets avec « Wireshark » en cliquant sur « l'aileron de requin ». Surfez sur le Web, par exemple sur le site du lycée Turgot, regardez une vidéo en ligne et ouvrez « Pronote ». Arrêtez la capture de paquets avec « wireshark » en cliquant sur le carré rouge.

Filtrez les adresses IP en saisissant « ip.addr == » suivi de l'adresse IP de votre ordinateur dans le champ suivant de « Wireshark » :

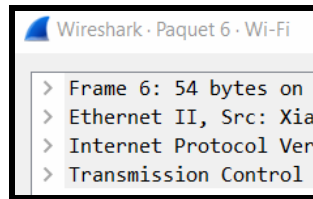


« ip.addr == » permettra de filtrer toutes les adresses sources et destinations concernées.

« ip.src == » permettra de filtrer toutes les adresses sources concernées.

« ip.dst == » permettra de filtrer toutes les adresses destinations concernées.

Dans la liste filtrée, afin d'observer plus précisément une trame, faites 2 clics gauches sur la première trame ayant le protocole « TCP » et une longueur non nulle. Une fenêtre similaire à la suivante s'ouvre :



Un clic sur « > » permettra de dérouler les informations relatives aux 4 couches (« Accès au réseau », « Internet », « Transport » et « Application ») du modèle TCP/IP de la trame et un sur « v » permettra de le réduire.

Complétez les données suivantes :

- Protocole de contrôle de transmissions :
 - Index de flux : ____
 - Port source : ____
 - Port destination : ____
 - Adresse IP source : ____
 - Adresse IP destination : ____
 - Avec l'outil « Whois » de « ping.eu » recherchez le nom et les coordonnées du détenteur de cette adresse IP : ____

 - Drapeaux actifs : ____

 - Longueur de l'entête : ____
 - Longueur du segment : ____
- Protocole Internet :
 - Longueur de l'entête : ____
 - Longueur total du paquet : ____
 - Drapeaux actifs : ____

- Ethernet :
 - Adresse MAC source : ____
 - Adresse MAC destination : ____
 - Type d'adresse IP : IPv ____
- Trame :
 - Longueur de la trame : ____

adresse MAC de : ____
adresse MAC de : ____

4.2. NOTION DE RESEAU (rappels de classe de première)

Un **réseau** est un ensemble d'objets interconnectés les uns avec les autres via une interface réseau. Il permet de faire circuler des éléments entre chacun de ces objets selon des règles bien définies.

Une adresse IPv4 d'une machine connectée à Internet se compose de deux parties :

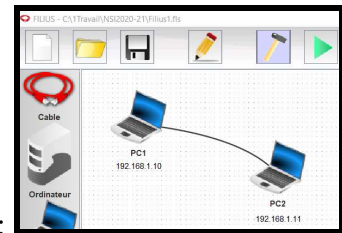
- L'adresse du réseau auquel appartient cette machine,
- Le numéro de poste de la machine (appelée aussi **hôte**) dans ce réseau.


Soit une machine dont l'adresse IP est 192.168.1.10,

- Si les deux premiers octets sont utilisés pour l'adresse du réseau qui est 192.168.0.0, la machine hôte est celle ayant pour numéro 1.10 dans ce réseau.
- Si les trois premiers octets sont utilisés pour l'adresse du réseau qui est 192.168. 1.0, la machine hôte est celle ayant pour numéro 10 dans ce réseau.

Installer « Filius » à partir du lien <https://www.lernsoftware-filius.de/Herunterladen>.
Attention à choisir la langue « Français ».

En « Mode conception » , dessinez le réseau suivant :




Les 2 ordinateurs appartiennent au même réseau. Le « Mode documentation »  permet d'écrire les adresses IP près des 2 ordinateurs.

En « Mode conception », en faisant un clic droit sur chaque ordinateur du réseau, configurez les ainsi :

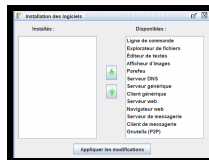
Nom	PC1	<input type="checkbox"/> Utiliser l'adresse IP comme nom
Adresse MAC	97:FE:1E:28:51:D1	<input type="checkbox"/> Activer routage des paquets
Adresse IP	192.168.1.10	<input type="checkbox"/> Adressage automatique par serveur DHCP
Masque	255.255.255.0	<input type="button" value="Configuration du service DHCP"/>
Passerelle		
Serveur DNS		

Nom	PC2	<input type="checkbox"/> Utiliser l'adresse IP comme nom
Adresse MAC	3D:DF:2F:43:29:EA	<input type="checkbox"/> Activer routage des paquets
Adresse IP	192.168.1.11	<input type="checkbox"/> Adressage automatique par serveur DHCP
Masque	255.255.255.0	<input type="button" value="Configuration du service DHCP"/>
Passerelle		
Serveur DNS		

En « Mode simulation » , avec un clic droit sur l'ordinateur PC1 et sélectionnez « Afficher le bureau » :




Faites un clic gauche sur « Installation des logiciels » pour obtenir cette



fenêtre : . Sélectionnez avec 2 clics « ligne de commande » puis cliquez sur



« Appliquer les modifications » pour obtenir l'affichage suivant . Cliquez sur l'icône « Ligne de commande » puis dans l'invité de commande saisir : « ping » suivi de l'adresse IP du deuxième ordinateur (192.168.1.11).

Validez et observez ce qu'il se passe. Quel est le but d'un « ping » ? :

Nous allons essayer de connecter 3 ordinateurs. Modifiez le réseau précédent pour y parvenir. Que remarquez-vous ?

Quel élément faut-il ajouter au réseau pour pouvoir connecter n ordinateurs ?

Connectez les 3 ordinateurs puis afin de vérifier le bon fonctionnement de votre réseau LAN, faites des « ping » depuis chaque ordinateur vers les autres ordinateurs.

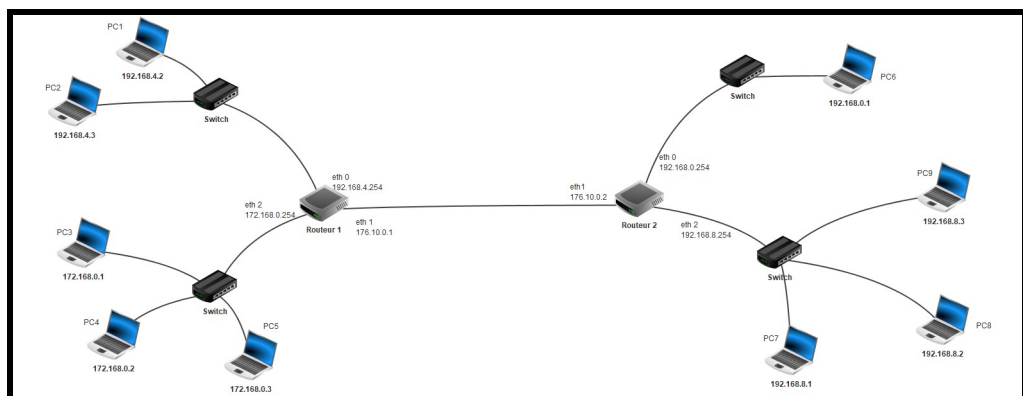
Nous allons essayer de connecter 2 réseaux LAN composés de 3 ordinateurs chacun. Visualisez la vidéo sur l'ENT : « 02_filius_man.mp4 » puis modifiez le réseau précédent pour y parvenir.

La **passerelle** (gateway) est l'adresse de l'élément qui va permettre de sortir du sous – réseau pour accéder aux autres.

Complétez le tableau suivant :

Noms	Adresses MAC	Adresses IP	Masques	Passerelles
PC1				
PC2				
PC3				
PC4				
PC5				
PC6				

Quelle est la particularité des adresses IP d'un même sous - réseau ?



Sur le schéma précédant :

- Entourez les différents sous - réseaux.
- Donnez le nom de ces sous - réseaux.

Pour extraire facilement le groupe de bits correspondant à l'adresse IPv4 d'un sous - réseau, on utilise un **masque de sous - réseau** (subnet mask, netmask ou address mask).

L'adresse du sous - réseau est obtenue en appliquant l'opérateur ET binaire entre l'adresse IPv4 et le masque de sous - réseau.

L'adresse de l'hôte à l'intérieur du sous - réseau est obtenue en appliquant l'opérateur ET entre l'adresse IPv4 et le complément à un du masque.

Exemples

Adresse IP de l'hôte : 192.168.1.10
Masque de sous – réseau : 255.255.255.0

En décimale :

192.168.1.10	&	255.255.255.0	=	192.168.1.0
192.168.1.10	&	0.0.0.255	=	0.0.0.10

En binaire :

	11000000.10101000.00000001.00001010	
&	11111111.11111111.11111111.00000000	
=	11000000.10101000.00000001.00000000	est l'adresse du sous - réseau

	11000000.10101000.00000001.00001010	
&	00000000.00000000.00000000.11111111	
=	00000000.00000000.00000000.00001010	est l'adresse de l'hôte dans le sous - réseau

Une notation plus courte du masque de sous – réseau nommée CIDR (Classless Inter-Domain Routing : routage inter-domaine sans classe). Elle donne le numéro du réseau suivi par une barre oblique (slash, « / ») et le nombre de bits gauche à 1 dans la notation binaire du masque de sous - réseau :

- Une adresse IP se terminant par /8, a un masque de sous – réseau « 255.0.0.0 ».
- Une adresse IP qui se termine par /16, a un masque de sous – réseau « 255.255.0.0 ».
- Une adresse IP qui se termine par /24, a un masque de sous – réseau « 255.255.255.0 ».

Soit un ordinateur ayant pour adresse IP 192.168.1.10/16, donnez :

- L'adresse IP de l'ordinateur :
- Le masque du sous - réseau auquel il appartient :
- L'adresse IP du sous - réseau auquel il appartient :
- La première adresse IP attribuable à un ordinateur de ce sous – réseau :
- La dernière adresse IP attribuable un ordinateur de ce sous – réseau :
- Le nombre d'adresse IP hôtes disponibles sur ce réseau :
- L'adresse IP de la passerelle du sous – réseau auquel il appartient :
- L'adresse IP de broadcast du sous – réseau auquel il appartient :

Le **Broadcast** (diffuser) est une méthode de transmission de données à l'ensemble des machines d'un réseau sans nécessité de réponse. L'adresse IP du broadcast est dernière adresse IP du réseau.

Soit un ordinateur ayant pour adresse IP 149.236.23.47/16, donnez :

- L'adresse IP de l'ordinateur :
- Le masque du sous - réseau auquel il appartient :
- L'adresse IP du sous - réseau auquel il appartient :

Complétez le tableau suivant :

IP notation CIDR	Adresses IP	Masques du sous - réseau	Adresse IP du réseau
201.34.156.75/24			
201.34.156.75/26			
201.34.156.75/20			

Soit la notation CIDR « /28 ».

- Quel est le masque de sous - réseau correspondant à cette notation ?
- Combien d'adresses différentes peut-on écrire dans un même réseau ?
- Pourquoi le nombre d'hôtes est limité à 14 dans ce réseau ?

Un fournisseur d'accès Internet se voit allouer un bloc d'adresses IP avec un CIDR de 19.

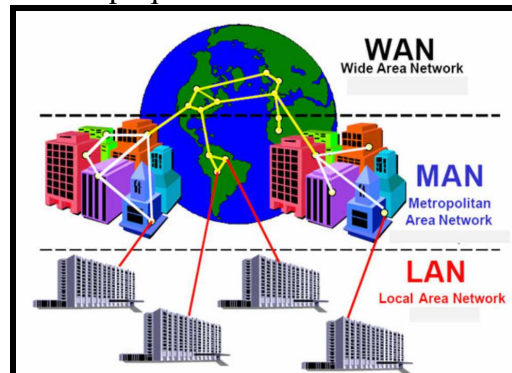
- Combien d'adresses IP différentes peut octroyer ce fournisseur d'accès ?
- Pourquoi seulement 8190 machines peuvent se voir attribuer une adresse IP différentes dans ce réseau ?

Vu les besoins de ses clients locaux, le fournisseur d'accès décide de créer des sous - réseaux de taille variable. Un client à besoin d'un réseau local de 200 ordinateurs et les autres ont besoin de liens pair - à - pair. Le CIDR attaché aux adresses IP est alors « /30 ».

- Quelle est la valeur maximale du nombre entier apparaissant dans le CIDR lié au réseau local de 200 ordinateurs ?
- Combien de machines différentes peut comporter un sous - réseau dont le CIDR est « /30 » ?

Il existe quatre types de réseaux :

- **PAN** (Personal Area Network : réseau personnel ou domestique) réseau à échelle d'une maison ou d'un appartement.
- **LAN** (Local Area Network : réseau local) réseau à échelle d'un bâtiment avec Wi-Fi et câble Ethernet.
- **MAN** (Metropolitan Area Network : réseau métropolitain) réseau à échelle d'une ville avec modem et fils.
- **WAN** (Wide Area Network : réseau à grande distance) réseau à échelle d'un pays ou un groupe de pays avec fibre optique et satellites.



<https://www.iifa.fr/reseaux>

Visualisez la vidéo (grâce aux paramètres, vous pouvez si besoin ralentir la vitesse de lecture) : <https://www.youtube.com/watch?v=c0Xj09s5hYA>

4.3. MATERIEL RESEAU

4.3.1. LE HUB

Le hub (concentrateur) est l'équivalent de la multiprise en électricité, il répète les données reçues en les transmettant à toutes les machines connectées, ce qui réduit considérablement la bande passante. Le HUB agit seulement au niveau de la couche 1 du modèle OSI, c'est-à-dire qu'il ne voit que des bits. Si au moins deux machines transmettent en même temps il y a collision. Le hub pose d'importants problèmes de sécurité en facilitant l'écoute sur le réseau.

4.3.2. LE SWITCH

A la différence du hub, le switch (commutateur) distribue les données à la machine destinataire, il travaille sur les deux premières couches « Physique » et « Liaison de données » du modèle OSI et sur la couche 1 « Accès réseau » du modèle TCP/IP. Le switch va « décoder » les entêtes de la trame pour trouver l'adresse MAC de destination. Il a une table de correspondance (adresse MAC = port x) et il renvoie la trame uniquement à ce destinataire. Le switch permet également d'éviter les collisions, si la machine que l'on tente de joindre est occupée, un nouvel essai sera effectué plus tard. Il n'y a donc plus de partage de bande passante et la sécurité est accrue.

4.3.3. LE ROUTEUR

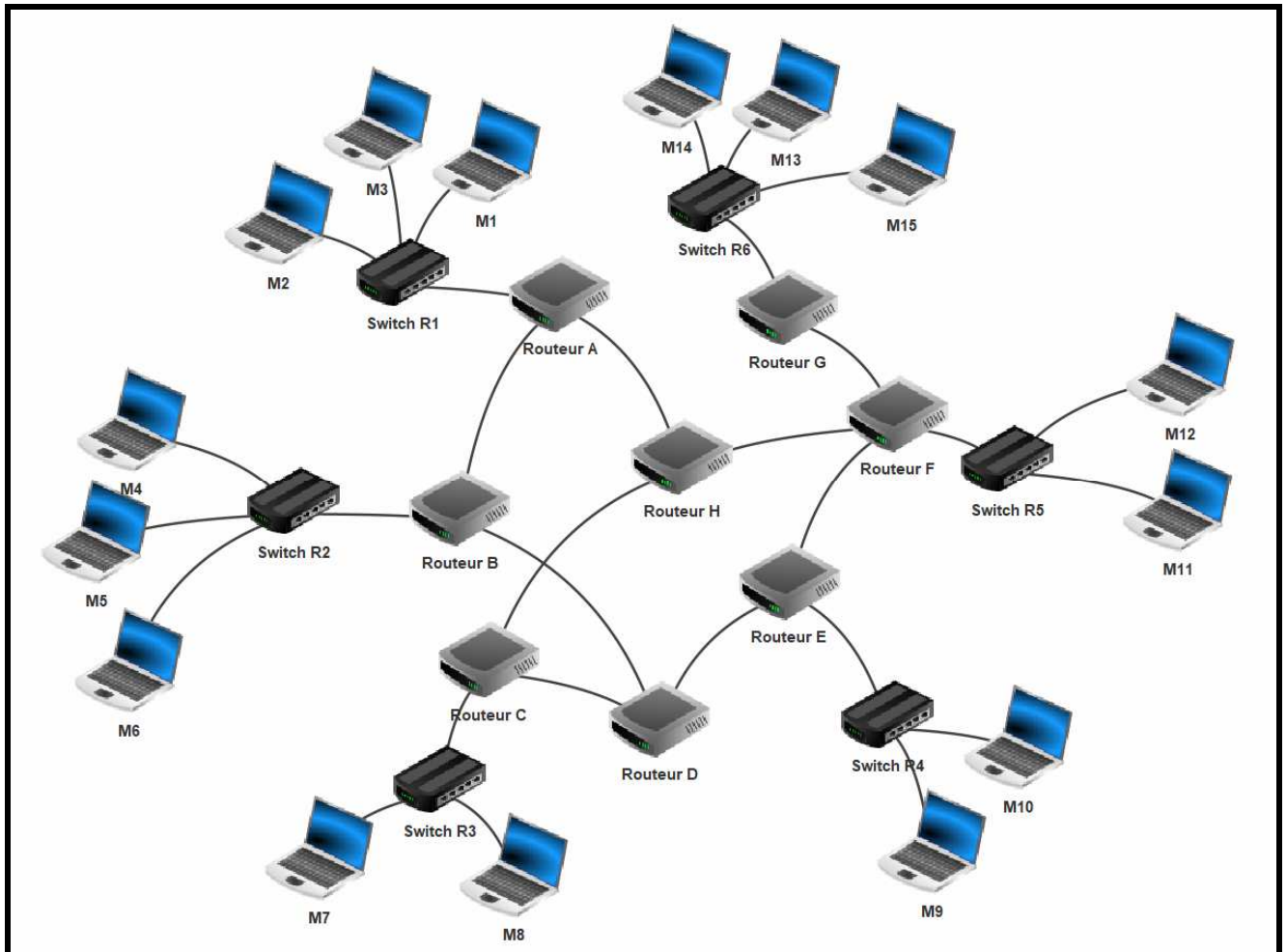
Les routeurs opèrent au niveau de la couche 3 « réseau » du modèle OSI. Le routeur permet de faire le lien entre différents sous - réseaux, il possède donc au moins deux interfaces réseaux (« cartes » réseau : Ethernet, Wifi, ...). Si plusieurs routes sont disponibles, il va choisir la meilleure pour acheminer les paquets. Il existe des routeurs capables de relier ensemble une dizaine de réseaux. N'importe quel ordinateur peut jouer le rôle de routeur dès lors qu'il possède au moins 2 interfaces réseau.

On distinguera :

- les routeurs d'accès qui permettent aux machines d'un sous - réseau d'accéder au réseau. On y connecte les clients et les serveurs, au sein de sous - réseaux (LAN).
- les routeurs internes qui constituent la structure du réseau lui-même. Leur rôle est d'acheminer les paquets au sein du réseau en se les relayant. Les distances entre eux sont généralement longues et nécessitent des technologies adaptées (fibre optique, satellites, ...).

Le routeur est un composant de la couche 2 « Internet » du modèle TCP/IP : il décapsule / désencapsule les paquets jusqu'à pouvoir lire les adresses IP des expéditeurs et destinataires.





Sur le schéma du réseau ci-dessus, entourez les différents réseaux.

Nommez les routeurs d'accès :

Nommez les routeurs internes :

4.4. LE ROUTAGE

Thomas Tallavinaigrette veut envoyer une lettre de remerciement à son ami Pacôme Toullemonde, habitant à Liège, qui lui a prêté un manuel de NSI. Il rédige donc un courrier, le place dans une enveloppe sur laquelle il inscrit le nom, le prénom, le numéro, la rue, le code postal, la ville et enfin le pays. Puis il dépose le tout à la poste de Limoges.

L'agent de la poste ne connaît pas Pacôme, mais il sait que la lettre doit aller en Belgique.

Une fois arrivée en Belgique, l'agent de poste Belge ne connaît toujours pas Pacôme, mais il sait qu'il doit envoyer la lettre vers un nouveau centre de tri, celui de Liège.

Arrivée au centre de Liège, la lettre va au bureau de poste correspondant au quartier de Pacôme. Et enfin elle sera remise au facteur qui distribue le courrier dans la rue de Pacôme, afin qu'il la dépose dans sa boîte aux lettres.

Nous pouvons ici faire une analogie entre les réseaux postaux et les réseaux informatiques :

- Thomas et son ami sont des machines (ordinateurs).
- Les centres de tris sont des routeurs.
- Le facteur est un switch.

4.4.1. DEFINITION

Le routage est le processus qui permet de sélectionner un chemin dans un ou plusieurs réseaux pour transmettre des données depuis un expéditeur jusqu'à un ou plusieurs destinataires. Le routage n'est pas exclusivement destiné aux réseaux informatiques tel qu'Internet, il existe aussi dans d'autres domaines comme les réseaux de transports, les réseaux téléphoniques, ...

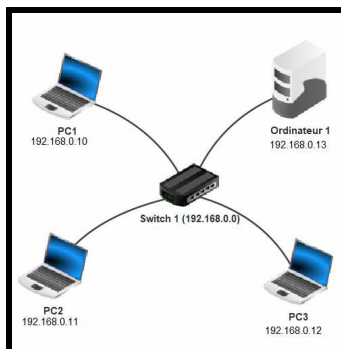
Il existe deux types de routage :

- Le **routage statique** qui consiste à indiquer l'adresse IP des réseaux que l'on cherche à atteindre. Les administrateurs vont configurer les routeurs un à un au sein du réseau afin d'y saisir les routes (Ils vont renseigner manuellement pour chaque adresse le nom de l'interface du routeur ou l'adresse IP du routeur voisin) à emprunter pour aller sur tel ou tel réseau. Chaque routeur ainsi paramétré permettra de faire le lien entre deux réseaux. Si le réseau global est complexe, la configuration peut être fastidieuse et source d'erreurs. De plus si le réseau évolue, il faudra le mettre à jour manuellement. Ce type de routage peut-être intéressant pour des raisons de sécurité si on veut maîtriser la route des paquets. Ce type de routage reste très localisé.
- Le **routage dynamique** permet quant à lui de se mettre à jour de façon automatique. Un protocole de routage va permettre aux différents routeurs de se comprendre et d'échanger des informations de façon périodique ou événementielle afin que chaque routeur soit au courant des évolutions du réseau sans aucune intervention de l'administrateur du réseau.


4.4.2. ROUTAGE ENTRE DEUX HOTES DU MEME SOUS - RESEAU

Dans un même sous - réseau, tous les hôtes doivent avoir la même adresse réseau (Net Id : identifiant réseau, par opposition à Host Id : identifiant hôte qui identifie la machine dans le réseau). Les paquets de données passent par les commutateurs (switch), un composant de la couche 1 « Accès réseau » du modèle TCP/IP . Un switch ne lit pas les adresses IP. Le routage se fait à l'aide des adresses MAC.

Avec « Filius » en « Mode conception », dessinez le réseau suivant (masque de réseau 255.255.255.0) :



En observant attentivement le schéma du réseau, en « Mode simulation », en réglant la

vitesse à 1%  , faites des « ping » depuis chaque ordinateur vers les autres ordinateurs afin de **décrire chronologiquement les échanges** :

En observant attentivement le schéma du réseau, relancer un « ping » depuis un ordinateur vers un autre (flèche du haut), et observer à nouveau attentivement le schéma du réseau afin de **décrire chronologiquement les échanges**. Quelle est la différence:

Cliquer sur le switch et relever dans la table d'allocation du switch : SAT (switch allocation table) à quels ordinateurs correspondent les adresses MAC qui s'y trouvent.

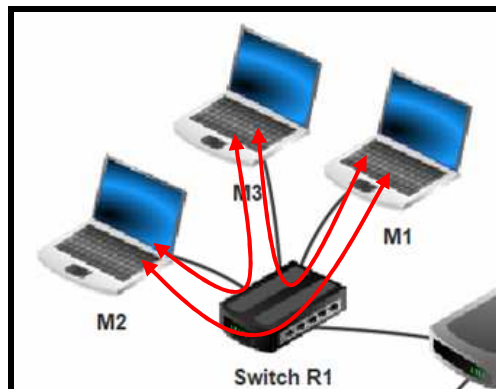
Faire un clic-droit sur tous les ordinateurs et choisir « Afficher les échanges de données ».

FILIUS permet d'afficher les détails de chaque message échangé (envoyé et reçu) pour chaque machine du réseau. Les détails sont séparés en fonction des couches OSI (Réseau et Internet).

En observant l'ensemble des échanges, sur tous les ordinateurs,

- Donner le nom du protocole qui a permis de remplir la table SAT du switch. A quelle couche appartient-il ?
- Quels ordinateurs ont reçu un message ARP ? Lequel a répondu ?
- Pourquoi dans certains messages on voit une adresse MAC valant FF:FF:FF:FF:FF:FF ?
- Conclure quant au comportement observé aux questions précédentes. En déduire le rôle de cette table SAT et comment le switch s'en sert.

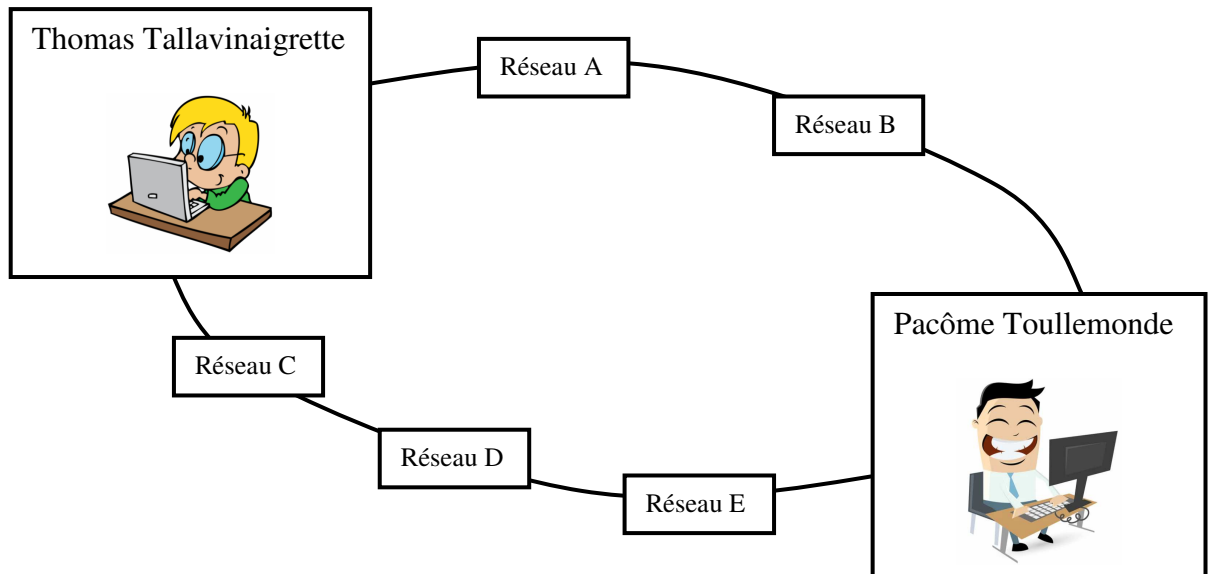
En revenant au réseau de la page 57, du fichier nommé « routage_simulation_6reseaux.flx », faites le même TP que le précédant sur les machines M1 à M3 et sur le Switch R1 :



4.4.3. ROUTAGE ENTRE DEUX HOTES DE SOUS – RESEAUX DIFFERENTS

4.4.3.1. QUEL EST LE MEILLEUR CHEMIN ?

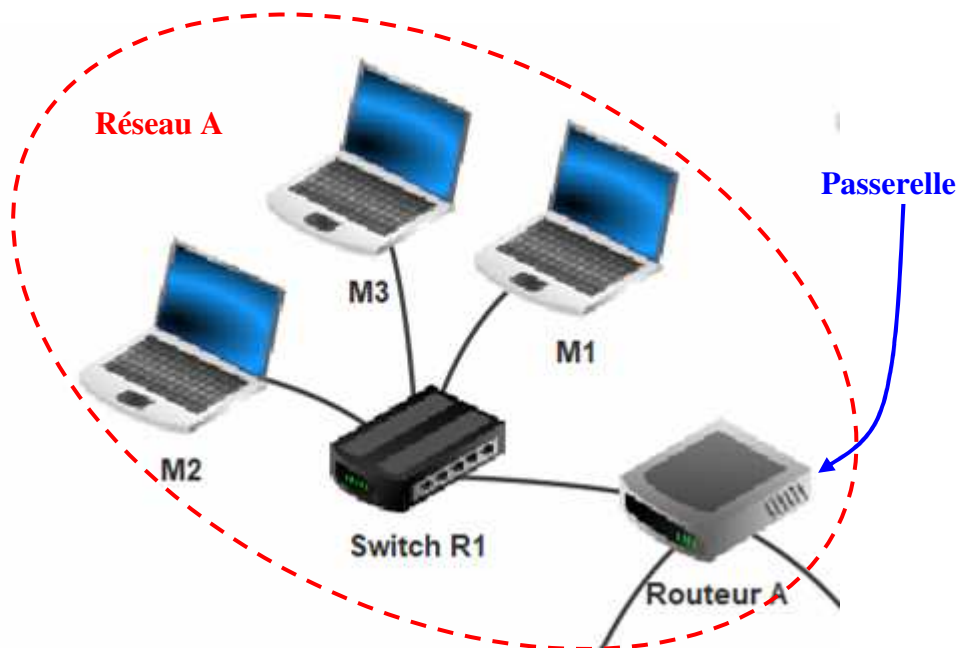
La plupart du temps, il y a plusieurs « routes » possibles et le chemin que l'on pense le plus « court » n'est pas toujours le plus rapide :

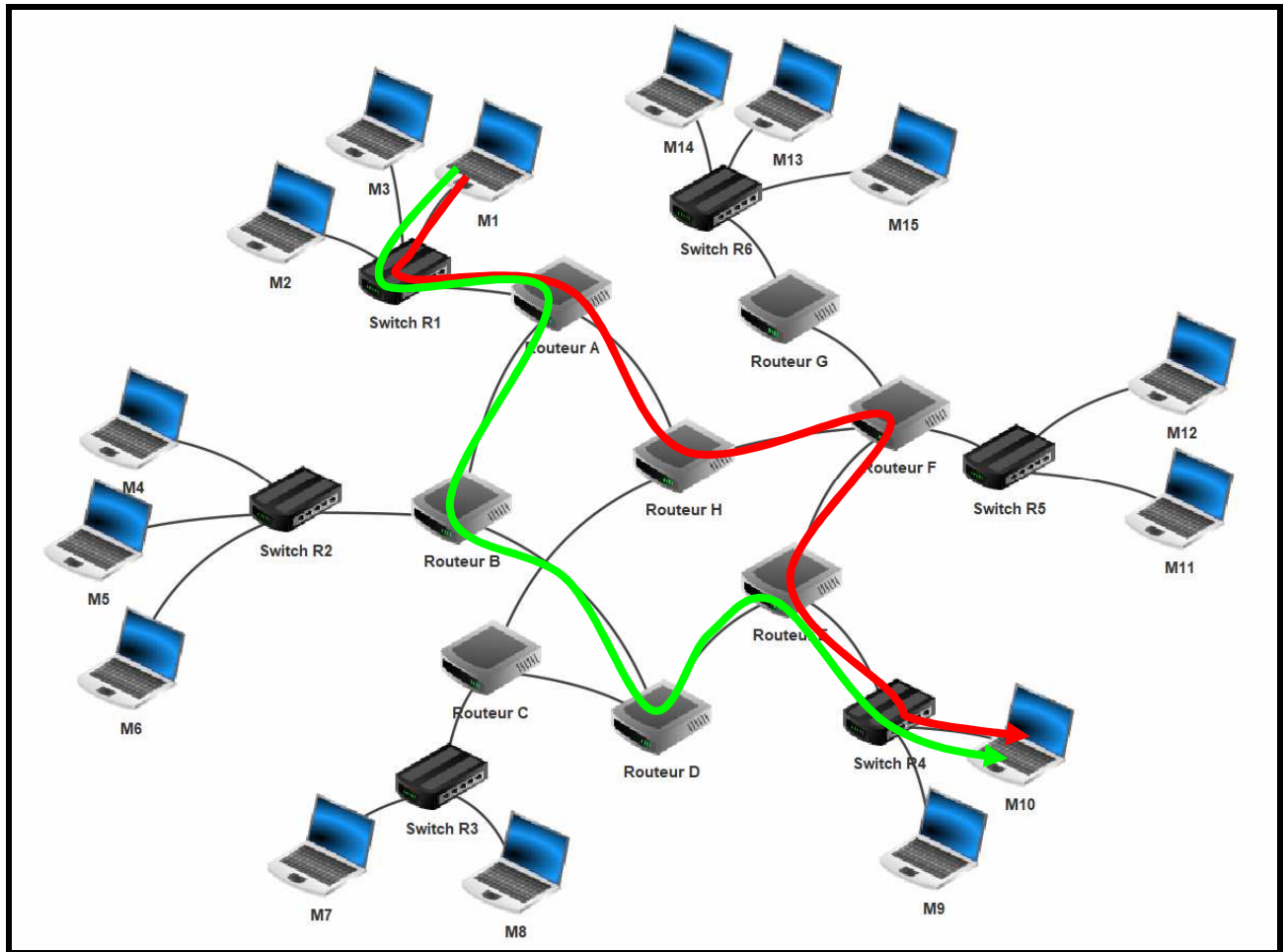


4.4.3.2. EN PRATIQUE

Deux hôtes de deux sous - réseaux différents ne possèdent pas la même adresse de réseau. Le destinataire n'étant pas sur le même réseau que l'expéditeur, le switch s'adresse à un routeur d'accès qui joue alors le rôle de passerelle (gateway). Chaque routeur du réseau possède une certaine « connaissance » de la topologie du réseau et peut faire passer le paquet qu'il reçoit à un routeur voisin.

En revenant au réseau de la page 57 :





La plupart du temps, il y a plusieurs « routes » possibles :

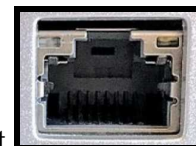
Déterminer trois chemins différents possibles permettant d'établir une connexion entre la machine M4 et M14.

4.4.3.3. TABLE DE ROUTAGE

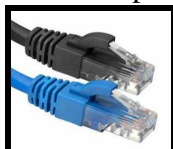
Chaque machine connectée à un réseau, et jouant un rôle de routeur, possède une table de routage : un tableau qui contient des informations permettant au routeur d'envoyer le paquet de données dans la « bonne direction ».

Un routeur possède plusieurs interfaces réseau (eth0, eth1, wlan0, wlan1, ...), par lequel il est connecté à des sous - réseaux (autre routeur, réseau local, ...) au sein desquels il possède une adresse IP.

WLAN (Wireless Local Area Network : réseau local sans fil) réseau sans fil à échelle d'un bâtiment. Wi-Fi est une marque de commerce qui peut être utilisé sur les appareils qui répondent aux normes 802.11.



Un routeur possède généralement plusieurs ports **Ethernet** pour accueillir plusieurs périphériques câblés sur un réseau. Avec des cables RJ45



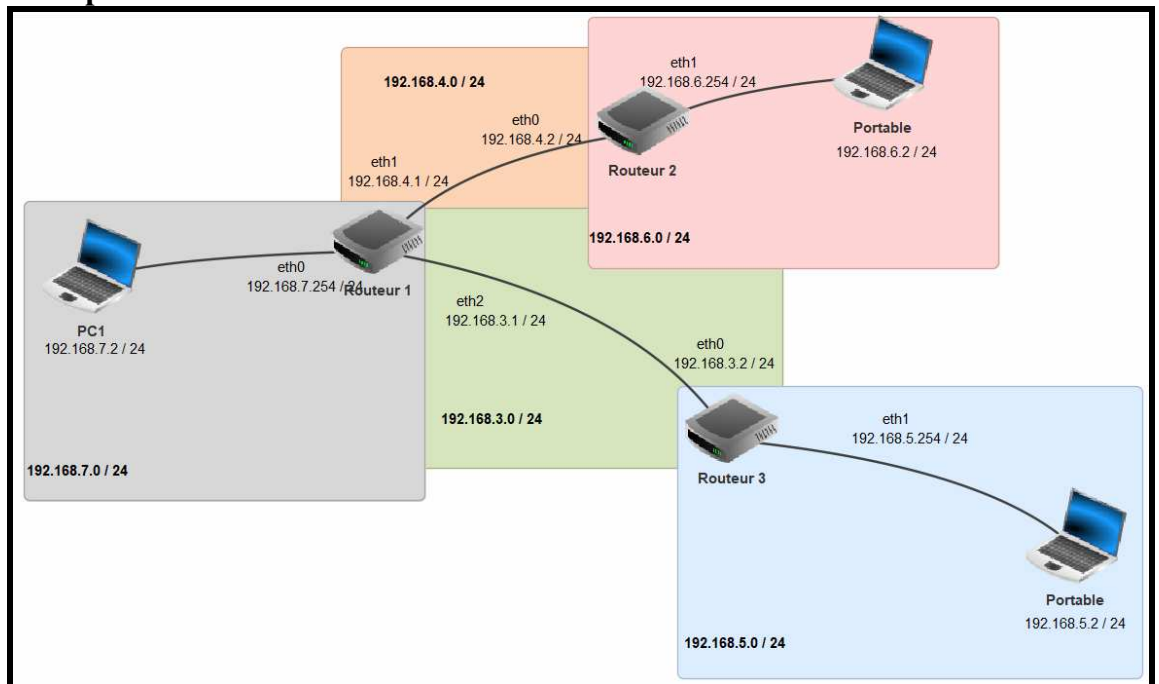
Chaque OS (systèmes d'exploitation) possède sa propre nomenclature pour les interfaces réseau :

- Nomenclature des interfaces réseau sous Linux :
<https://www.linuxtricks.fr/wiki/nomenclature-des-interfaces-reseau-sous-linux-enp0s3-wlo1>
- Nomenclature des interfaces réseau sous Windows :
<https://www.linuxtricks.fr/wiki/windows-lignes-de-commandes-utiles>

On peut obtenir la liste des interfaces disponibles sur une machine à l'aide de commandes :

- Linux : « ifconfig »
- Windows : « ipconfig »

Exemple :



Le « routeur 1 » ci-dessus possède 3 interfaces Ethernet (eth0, eth1 et eth2), lui permettant de faire le lien entre 3 réseaux.

Dans la **table de routage** d'un routeur, on trouve les informations suivantes :

- Les adresses IP du routeur (il en possède une pour chacune de ses interfaces)
- Les adresses IP des sous - réseaux auquel il est connecté
- Les routes (directions à prendre) qu'il faut suivre pour atteindre un réseau
 - route par défaut (il en faut bien une)
 - routes statiques (configurées explicitement par l'administrateur)
 - routes dynamiques (appries par des protocoles de routage dynamique)

Chaque ligne de la table contient une **route** :

- l'adresse du réseau de **destination**
- la **direction** à prendre pour l'atteindre :
 - **Interface** de sortie du routeur (adresse du routeur sur le sous - réseau de sortie)
 - **Passerelle** (adresse de la prochaine machine sur le sous - réseau de sortie)

- la **métrique** correspond au nombre de sauts IP (hop = saut : saut de routeur à routeur) nécessaires pour atteindre le réseau destination : Un ordinateur directement connecté sur un routeur aura une métrique de 0 et un routeur directement connecté sur un autre routeur aura une métrique de 1. **Attention dans certains exercices, la métrique correspond à la « distance » à parcourir** : Une machine (ordinateur ou routeur) directement connectée sur un routeur aura une métrique de 1. Dans la suite, nous utilisons la métrique correspondant au nombre de sauts IP.

Lorsqu'un routeur reçoit un paquet, il récupère l'adresse du réseau de destination :

- si cette adresse est dans sa table de routage, il envoie le paquet vers l'interface associée,
- dans le cas contraire le paquet est envoyé via l'interface par défaut.

Exemples :

Si le « routeur 1 » reçoit un paquet à destination de l'ordinateur 192.168.7.2 / 24, sa table de routage pourra contenir la route suivante :

Réseau destination	Passerelle	Interface	Métrique
192.168.7.0 / 24		eth0	0

L'ordinateur est directement relié au routeur, il n'y a donc pas besoin de passerelle pour l'atteindre. La métrique est de 0 (aucun « saut » pour l'atteindre)

Si le « routeur 1 » reçoit un paquet à destination de l'hôte 192.168.5.2 / 24, sa table de routage pourra contenir la route suivante :

Réseau destination	Passerelle	Interface	Métrique
192.168.5.0 / 24	192.168.3.2	eth2	1

L'ordinateur est relié via le port Ethernet 2 par l'intermédiaire du routeur 3, la passerelle pour l'atteindre est l'adresse IP entrante de ce routeur. La métrique est de 1 (1 « saut » ou 1 changement de réseau pour l'atteindre)

Si le « routeur 1 » reçoit un paquet à destination de l'hôte 192.168.6.2 / 24, sa table de routage pourra contenir la route suivante :

Réseau destination	Passerelle	Interface	Métrique
192.168.6.0 / 24	192.168.4.2	eth1	1

L'ordinateur est relié via le port Ethernet 1 par l'intermédiaire du routeur 2, la passerelle pour l'atteindre est l'adresse IP entrante de ce routeur. La métrique est de 1 (1 « saut » ou 1 changement de réseau pour l'atteindre)

Si le « routeur 1 » reçoit un paquet à destination d'un réseau qui n'est pas dans sa table, il choisit la route par défaut :

Réseau destination	Passerelle	Interface	Métrique
Défaut ou 0.0.0.0 / 0	192.168.4.2	eth1	1
Défaut ou 0.0.0.0 / 0	192.168.2.2	eth2	1

Visualisez la vidéo jusqu'à 17'06'' (grâce aux paramètres, vous pouvez si besoin ralentir la vitesse de lecture) : <https://www.youtube.com/watch?v=ZqW4nOkDBv4>

En pratique une table de routage ne comporte que des adresses IP :

- L'adresse du réseau de destination n'est pas au format CIRC mais au format IP réseau + masque réseau : 192.168.7.0 / 24 donnera 192.168.7.0 et 255.255.255.0

- L'interface est décrite par la propre adresse IP du routeur dans le sous - réseau auquel il est connecté avec cette interface : eth0 donnera 192.168.7.254

La table de routage du « routeur 1 » est donc :

Réseau destination	Passerelle	Interface	Métrique
192.168.7.0 / 24	-	eth0	0
192.168.4.0 / 24	-	eth1	0
192.168.3.0 / 24	-	eth2	0
192.168.6.0 / 24	192.168.4.2	(eth1)	1
192.168.5.0 / 24	192.168.3.2	(eth2)	1

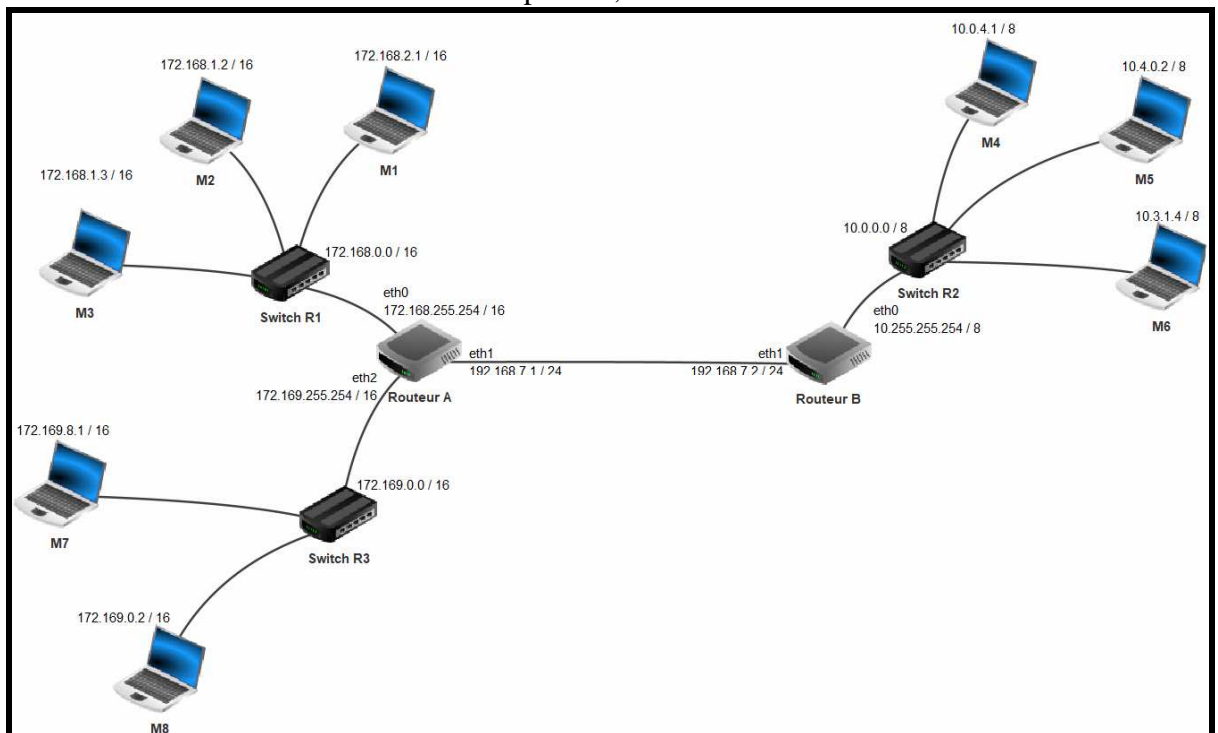
Complétez la table de routage du « routeur 2 » :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur 3 » :

Réseau destination	Passerelle	Interface	Métrique

Avec « Filius » en « Mode conception », dessinez le réseau suivant :



Afin de vérifier le bon fonctionnement de votre réseau, avec « Filius » en mode « simulation », faites des « ping » depuis chaque ordinateur vers les autres ordinateurs.

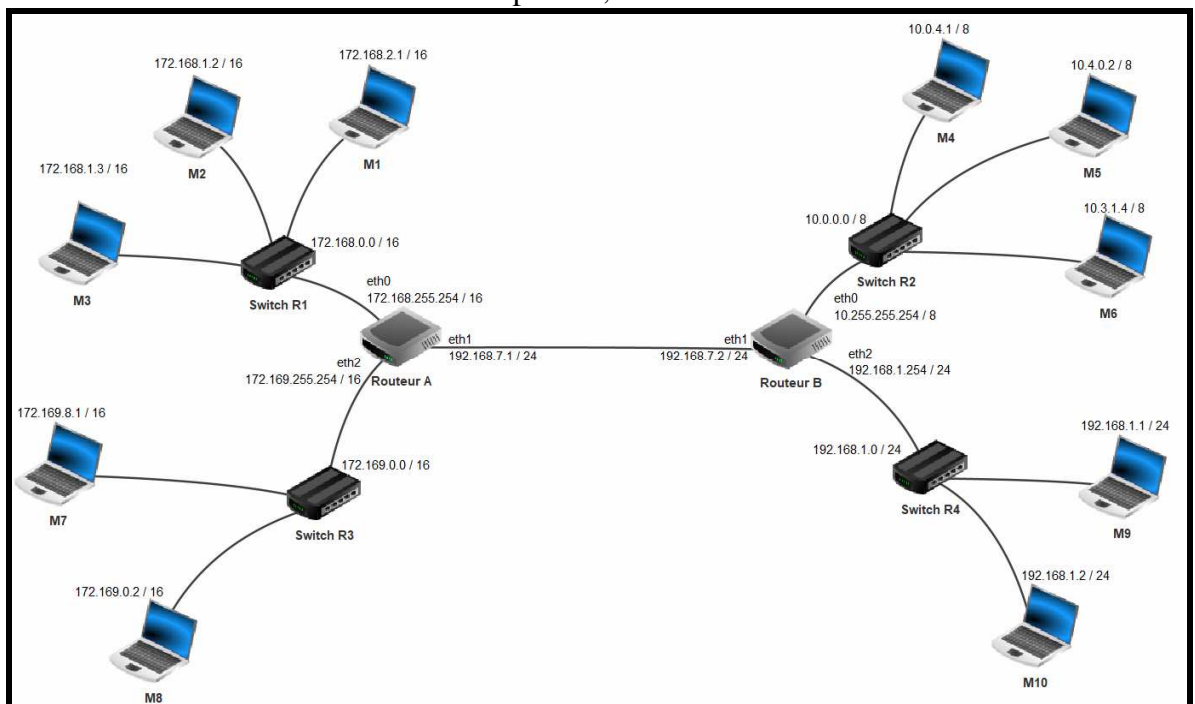
Complétez la table de routage du « routeur A » :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur B » :

Réseau destination	Passerelle	Interface	Métrique

Avec « Filius » en « Mode conception », dessinez le réseau suivant :



Afin de vérifier le bon fonctionnement de votre réseau, avec « Filius » en mode « simulation », faites des « ping » depuis chaque ordinateur vers les autres ordinateurs.

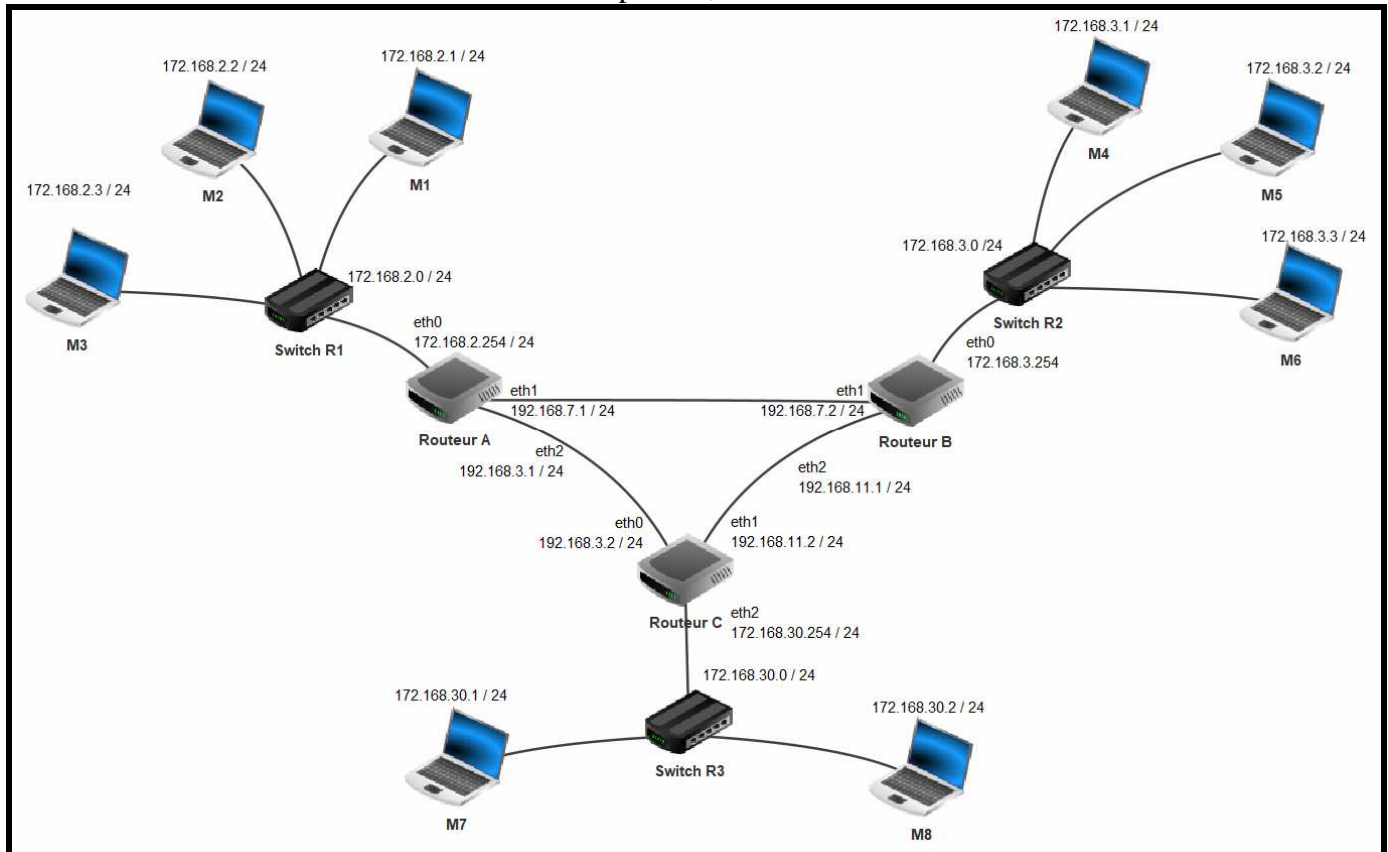
Complétez la table de routage du « routeur A » :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur B » :

Réseau destination	Passerelle	Interface	Métrieque

Avec « Filius » en « Mode conception », dessinez le réseau suivant :



Afin de vérifier le bon fonctionnement de votre réseau, avec « Filius » en mode « simulation », faites des « ping » depuis chaque ordinateur vers les autres ordinateurs.

Complétez la table de routage du « routeur A » :

Réseau destination	Passerelle	Interface	Métrieque

Complétez la table de routage du « routeur B » :

Réseau destination	Passerelle	Interface	Métrieque

Complétez la table de routage du « routeur C » :

Réseau destination	Passerelle	Interface	Métrieque

Que remarquez-vous ?

Dans des réseaux très complexes, chaque routeur aura une table de routage qui comportera de très nombreuses lignes (des dizaines voir des centaines...). En effet chaque routeur devra savoir vers quelle interface réseau il faudra envoyer un paquet afin qu'il puisse atteindre sa destination. On peut trouver dans une table de routage plusieurs lignes pour une même destination, il peut en effet, à partir d'un routeur donné, exister plusieurs chemins possibles pour atteindre la destination. Dans le cas où il existe plusieurs chemins possibles pour atteindre la même destination, le routeur va choisir le « chemin le plus court ». Pour choisir ce « chemin le plus court », le routeur va utiliser la métrieque : plus la valeur de la métrieque est petite, plus le chemin pour atteindre le réseau est « court ». Un réseau directement lié à un routeur aura une métrieque de 0.

Comme nous l'avons déjà vu ci-dessus, il existe 2 méthodes :

- le routage statique : chaque ligne doit être renseignée « à la main ». Cette solution est seulement envisageable pour des très petits réseaux de réseaux
- le routage dynamique : tout se fait « automatiquement », on utilise des protocoles qui vont permettre de « découvrir » les différentes routes automatiquement afin de pouvoir remplir la table de routage automatiquement.

Exercice :

Soit un réseau constitué de 6 routeurs A, B, C, D, E et F, dont les tables de routage sont :

Routeur A		Routeur B		Routeur C		Routeur D		Routeur E		Routeur F	
Destination	Passerelle	Destination	Passerelle	Destination	Passerelle	Destination	Passerelle	Destination	Passerelle	Destination	Passerelle
B	B	A	A	A	A	A	E	A	B	A	E
C	C	C	C	B	B	B	E	B	B	B	E
D	B	D	E	D	B	C	E	C	B	C	E
E	B	E	E	E	B	E	E	D	D	D	D
F	B	F	E	F	B	F	F	F	F	E	E

Dessinez une représentation sous forme de graphe du réseau ainsi constitué.

4.4.4. PROTOCOLES DE ROUTAGE RIP ET OSPF

Un réseau de réseaux comportant des routeurs peut être modélisé par un graphe. Chaque routeur est un sommet et chaque liaison entre deux routeurs ou entre un routeur et un switch est une arête. Les algorithmes utilisés par les protocoles de routages sont donc des algorithmes issus de la théorie de graphes. On trouve plusieurs protocoles de routage, nous allons en voir deux : RIP (Routing Information Protocol : Protocole d'informations de routage) et OSPF (Open Shortest Path First : Ouvrez le chemin le plus court en premier).

Visualisez les 2 vidéos pour comprendre les protocoles de routage (grâce aux paramètres, vous pouvez si besoin ralentir la vitesse de lecture) :

<https://www.youtube.com/watch?v=sT9-lcbjqzI>

<https://www.youtube.com/watch?v=bslQLLzzxcg>

4.4.4.1. PROTOCOLE DE ROUTAGE RIP

Le protocole RIP (Routing Information Protocol) est un des premiers algorithmes utilisés pour établir les tables de routage des routeurs interconnectant des réseaux. Il s'appuie sur l'algorithme de Bellman-Ford (algorithme qui permet de calculer les plus courts chemins dans un graphe).

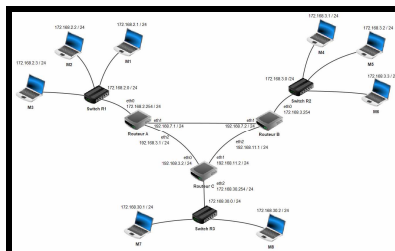
Il comporte les 3 règles suivantes :

- 1 Initialement (à la mise sous tension), la table de routage d'un routeur contient uniquement des informations sur ses voisins directs (ceux auxquels il est connecté).
- 2 Chaque routeur envoie périodiquement (toutes les 30 secondes), à tous ses voisins (routeurs adjacents), un **message** contenant :
 - la liste de toutes les **adresses de réseaux** qu'il connaît (ses voisins directs et ceux qu'il a reçus auparavant)
 - leur **métrique** (le nombre de sauts pour les atteindre)
 Ce couple d'informations (réseau, métrique) est appelé **vecteur de distance**, d'où le nom de ce protocole de routage : **routage à vecteur de distance**.
- 3 À chaque réception d'un message de ce type, un routeur met à jour sa table de routage avec les informations reçues. Quatre cas peuvent se présenter :
 1. il découvre une nouvelle route vers un sous réseau qu'il ne connaissait pas encore et il l'ajoute à sa table
 2. il découvre une nouvelle route plus courte vers un sous réseau qu'il connaissait déjà et il remplace l'ancienne route par la nouvelle
 3. il reçoit une nouvelle route plus longue vers un sous réseau qu'il connaissait déjà et il ne fait rien
 4. il reçoit une route existante dans sa table (passant par le même voisin), mais plus longue. Cela signifie que la route s'est allongée (panne ?). Il met sa table à jour

Visualisez la vidéo pour comprendre les protocoles de routage RIP (grâce aux paramètres, vous pouvez si besoin ralentir la vitesse de lecture) :

<https://www.youtube.com/watch?v=kzablGaqUXM>

Reprenons le réseau vu ci-dessus :



Complétez la table de routage du « routeur A » à la mise sous tension (étape 0) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur B » à la mise sous tension (étape 0) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur C » à la mise sous tension (étape 0) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur A » au bout de 30 secondes après la mise sous tension (étape 1) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur B » au bout de 30 secondes après la mise sous tension (étape 1) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur C » au bout de 30 secondes après la mise sous tension (étape 1) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur A au bout de 1 minute après la mise sous tension (étape 2) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur B au bout de 1 minute après la mise sous tension (étape 2) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur C au bout de 1 minute après la mise sous tension (étape 2) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur A après plus de 1 minute après la mise sous tension (étape 3) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur B après plus de 1 minute après la mise sous tension (étape 3) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur C après plus de 1 minute après la mise sous tension (étape 3) :

Réseau destination	Passerelle	Interface	Métrique

Que remarquez-vous par rapport aux tables de routage finales hors protocole RIP (pages 66 et 67) ?

Délai de convergence

On appelle délai de convergence le temps nécessaire à ce que l'ensemble des routeurs soient configurés pour offrir les meilleurs routes possible. À chaque modification du réseau (ajout ou suppression de routeurs), il faut un certain temps pour que les échanges de messages RIP mènent à une situation stable.

C'est pour limiter ce délai de convergence que le protocole RIP est **limité à 15 sauts**.

Détection des pannes

Lorsqu'un routeur ne reçoit pas de réponse de la part d'un autre routeur à une demande RIP (après un certain laps de temps, 3 minutes), il considère que le routeur en question est en panne.

Afin qu'aucun paquet ne soit plus dirigé dans cette direction, il prévient ses voisins en leur envoyant une métrique égale à 16 (plus grande valeur possible pour le protocole RIP ce qui correspond à une métrique infinie) concernant toutes les routes passant par le routeur qui ne répond pas.

Boucle de routage

Une boucle de routage est le phénomène qui se produit lorsqu'un paquet tourne en boucle et ne peut jamais atteindre sa destination (dans notre exemple ci-dessus, boucle formée par les routeurs A, B et C).

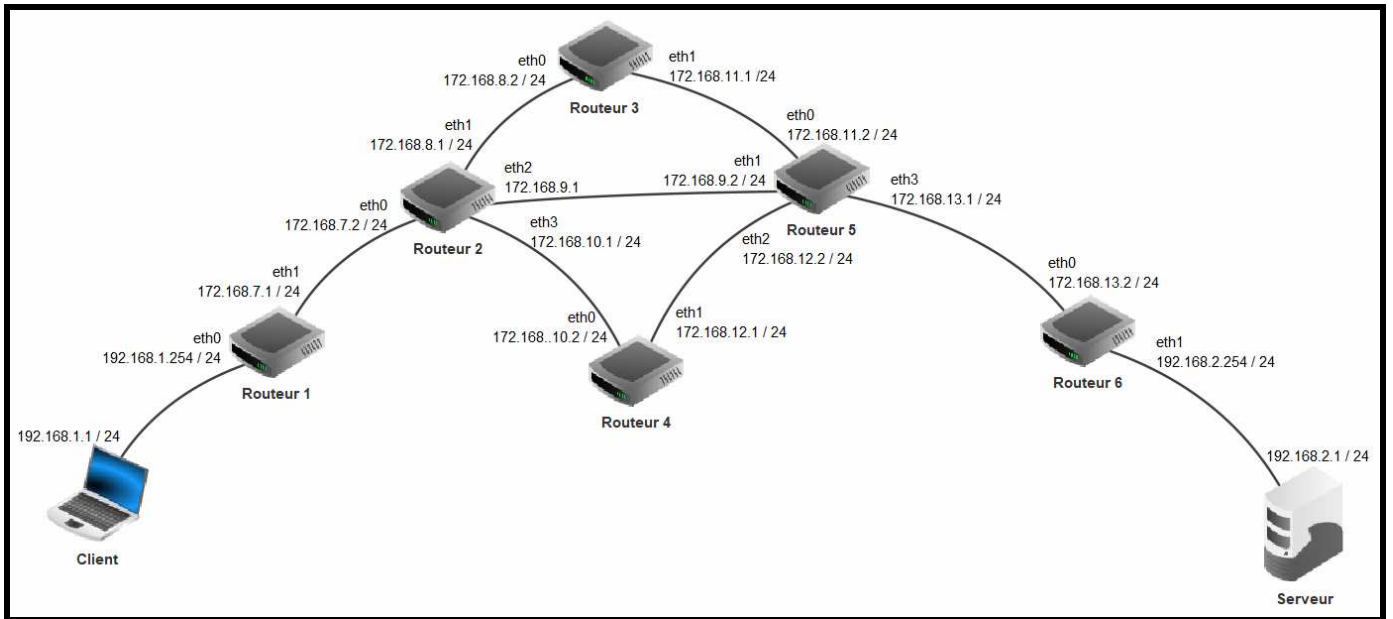
Différentes règles dans le protocole RIP permettent d'éviter que cela se produise :

- la règle « split horizon » (horizon partagé), qui interdit à un routeur d'envoyer une information de routage via le routeur qui la lui a envoyé.
- Au delà d'un nombre de 15 sauts, les paquets sont supprimés.

Le protocole RIP est aujourd'hui très rarement utilisé dans les grandes infrastructures car il génère, du fait de l'envoi périodique de messages, un trafic réseau important (surtout si les tables de routages contiennent beaucoup d'entrées). De plus, le protocole RIP est limité à 15 sauts (on traverse au maximum 15 routeurs pour atteindre sa destination). On lui préfère donc souvent le protocole OSPF.

4.4.4.1.1. UN AUTRE EXEMPLE DE PROTOCOLE DE ROUTAGE RIP

Avec « Filius » en « Mode conception », dessinez le réseau suivant. Puis afin de vérifier le bon fonctionnement de votre réseau, avec « Filius » en mode « simulation », faites des « ping » depuis chaque ordinateur vers l'autre ordinateur.



Complétez la table de routage du « routeur 1 » à la mise sous tension (étape 0) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur 2 » à la mise sous tension (étape 0) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur 3 » à la mise sous tension (étape 0) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur 1 » au bout de 30 secondes après la mise sous tension (étape 1) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur 2 » au bout de 30 secondes après la mise sous tension (étape 1) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur 3 » au bout de 30 secondes après la mise sous tension (étape 1) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur 1 » au bout de 1 minute après la mise sous tension (étape 2) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur 2 » au bout de 1 minute après la mise sous tension (étape 2) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur 3 » au bout de 1 minute après la mise sous tension (étape 2) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur 1 » après plus de 1 minute après la mise sous tension (étape 3) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur 2 » après plus de 1 minute après la mise sous tension (étape 3) :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur 3 » après plus de 1 minute après la mise sous tension (étape 3) :

Réseau destination	Passerelle	Interface	Métrique

Simulation d'une panne : soit le réseau précédant avec la liaison « 172.168.9.1 - 172.168.9.2 » Complétez la table de routage stabilisées (après plus de 1 minute après la mise sous tension) du « routeur 1 » selon le protocole RIP, puis conclure :

Réseau destination	Passerelle	Interface	Métrique

Avec « Filius » en mode « simulation », faites des « traceroute » depuis chaque ordinateur vers l'autre ordinateur.

Notez le résultat du « traceroute » fait depuis le « client » :

Notez le résultat du « traceroute » fait depuis le « Serveur » :

Avec « Filius » en « Mode conception », sur le réseau précédant supprimez la liaison « 172.168.9.1 - 172.168.9.2 ».

Avec « Filius » en mode « simulation », faites des « traceroute » depuis chaque ordinateur vers l'autre ordinateur, puis conclure.

Notez le résultat du « traceroute » fait depuis le « client » :

Notez le résultat du « traceroute » fait depuis le « Serveur » :

4.4.4.2. PROTOCOLE DE ROUTAGE OSPF

Pour le protocole de routage OSPF (Open Shortest Path First), comme pour le du protocole RIP, les routeurs échangent entre eux des informations, mais ces échanges sont plus « complets », ce qui permet de réduire l'occupation du réseau.

- Tous les routeurs ont une vision globale et identique du réseau : pour cela, ils reçoivent des informations depuis tout le réseau
- Les distances prennent en compte le nombre de routeur à traverser (nombre de sauts), mais également le débit binaire de chaque « liaison » (bande passante) exprimé en bits/s.

Le protocole OSPF permet à chaque routeur de connaître le graphe complet des liaisons entre tous les routeurs du réseau, avec leurs débits.

Le « meilleur » chemin n'est pas forcément le plus court, mais le plus rapide.

Le protocole OSPF s'appuie sur l'algorithme de Dijkstra afin de déterminer la meilleure route (en terme de débit) permettant d'atteindre la destination.

Visualisez les 2 vidéos pour comprendre l'algorithme de Dijkstra (grâce aux paramètres, vous pouvez si besoin ralentir la vitesse de lecture) :

<https://www.youtube.com/watch?v=MybdP4kice4>

<https://www.youtube.com/watch?v=rHylCtXtdNs>

La valeur de **métrique** retenue par le protocole OSPF n'est en réalité pas le débit, mais le **coût**. C'est un nombre entier : $\text{coût} = \frac{10^8}{\text{bande passante (en bits / s)}}$:

10^8 correspond au débit maximum de :

$100\,000\,000 \text{ bits / s} = 100 \text{ Mbps} = 100 \text{ mégabits / s}$

et où le débit est exprimé en bits par seconde.

Si la route possède plusieurs liaisons, le coût total de la route est la somme des coût des liaisons.

Le coût ne peut pas être nul. On arrondira à 1 tous les résultats inférieurs à 1.

Comme pour le protocole RIP, plus la métrique est petite, meilleure est la route.

Un ensemble de routeurs peut-être modélisé par un graphe pondéré où les sommets sont les routeurs et les poids les coûts.

Principe :

- Chaque routeur établit des relations d'adjacence avec ses voisins immédiats en envoyant des messages « hello » à intervalle régulier (il y a création d'une table de voisinage).
- Chaque routeur communique ensuite la liste des réseaux auxquels il est connecté par des messages Link-State Advertisements (LSA : annonces d'état de lien) propagés de proche en proche à tous les routeurs du réseau.
- L'ensemble des LSA forme une base de données de l'état des liens : Link-State DataBase (LSDB) pour chaque aire (regroupement de plusieurs routeurs), qui est identique pour tous les routeurs participants dans cette aire.
- Chaque routeur utilise ensuite l'algorithme de Dijkstra, Shortest Path First (SPF : chemin le plus court en premier) pour déterminer la route la plus rapide vers chacun des réseaux connus dans la LSDB.
- En cas de changement de topologie, de nouveaux LSA sont propagés de proche en proche, et l'algorithme SPF est exécuté à nouveau sur chaque routeur.

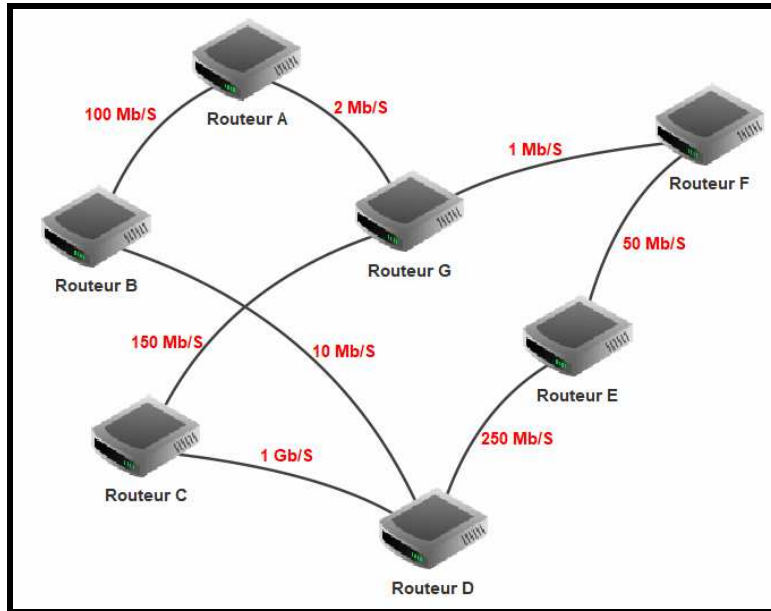
Ce protocole de routage est nommé **routage à état de lien**.

Visualisez la pour comprendre le protocole de routage OSPF (grâce aux paramètres, vous pouvez si besoin ralentir la vitesse de lecture) :

<https://www.youtube.com/watch?v=-utHPKREZV8>

<https://www.youtube.com/watch?v=FeZl3Xl7j84&t=26s>

Soit le réseau suivant dans lequel les débits sont indiqués sur les liaisons entre les routeurs :

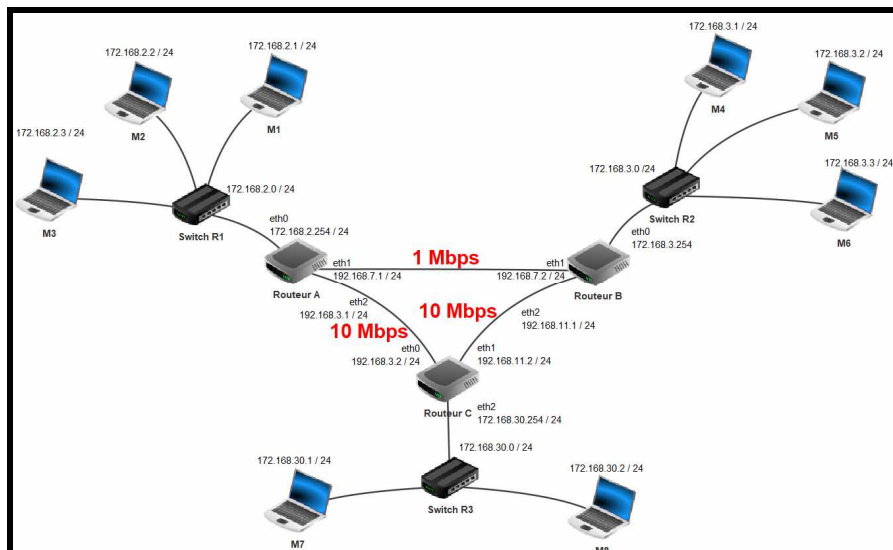


Indiquez le parcours entre les routeurs F et C avec le protocole de routage RIP :

Indiquez le parcours entre les routeurs F et C avec le protocole de routage OSPF :

Calculez les coûts des liaisons du réseau précédent et indiquez les sur les liaisons.

Soit le réseau suivant dans lequel les débits sont indiqués sur les liaisons entre les routeurs :



Indiquez le parcours entre les routeurs A et B avec le protocole de routage RIP :

Indiquez le parcours entre les routeurs A et B avec le protocole de routage OSPF :

Calculez les coûts des liaisons du réseau précédent et indiquez les sur les liaisons.

Selon le protocole OSPF, indiquez les « meilleurs » chemins (c'est à dire le chemin qui a le coût le plus faible) en partant des routeurs A, B et C pour atteindre les 2 autres routeurs :

Complétez la table de routage du « routeur A » suivante avec le protocole de routage OSPF :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur B » suivante avec le protocole de routage OSPF :

Réseau destination	Passerelle	Interface	Métrique

Complétez la table de routage du « routeur C » suivante avec le protocole de routage OSPF :

Réseau destination	Passerelle	Interface	Métrique

Sur ces 3 tables de routage précédentes, indiquez les « meilleurs » et les « pires » chemins.

Soit un réseau fonctionnant sous protocole de routage OSPF et composé de 7 routeurs (A à G). Le routeur A reçoit les informations suivantes :

- Liaison A - B avec un coût de 1
- Liaison A - C avec un coût de 1000
- Liaison A - D avec un coût de 100
- Liaison B - D avec un coût de 10
- Liaison C - E avec un coût de 200
- Liaison C - F avec un coût de 100
- Liaison D - E avec un coût de 1
- Liaison E - G avec un coût de 100
- Liaison F - G avec un coût de 10

Représentez le réseau sous forme d'un graphe où les sommets sont les routeurs et les arcs portent les coûts :

Calculez toutes les routes possibles entre les routeurs A et E et choisir celle qui présente le coût le plus faible. Quel est le coût de la liaison A-E ?

4.4.5. LE ROUTAGE A L'EPREUVE ECRITE**4.4.5.1.SUJET 1 , EXERCICE 2**

Un constructeur automobile utilise des ordinateurs pour la conception de ses véhicules. Ceux-ci sont munis d'un système d'exploitation ainsi que de nombreuses applications parmi lesquelles on peut citer :

- un logiciel de traitement de texte ;
- un tableur;
- un logiciel de Conception Assistée par Ordinateur (CAO);
- un système de gestion de base de donnée (SGBD).

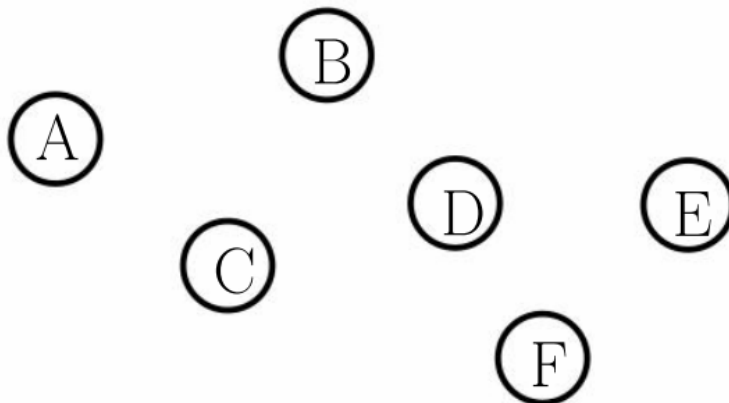
Chaque ordinateur est équipé des périphériques classiques : clavier, souris, écran et est relié à une imprimante réseau.

1. Ce constructeur automobile possède six sites de production qui échangent des documents entre eux. Les sites de production sont reliés entre eux par six routeurs A , B , C, D, E et F. On donne ci dessous les tables de routage des routeurs A à F obtenus avec le protocole RIP :

Routeur A		Routeur B		Routeur C		Routeur D		Routeur E		Routeur F	
Destination	Passerelle	Destination	Passerelle	Destination	Passerelle	Destination	Passerelle	Destination	Passerelle	Destination	Passerelle
B	B	A	A	A	A	A	C	A	B	A	D
C	C	C	C	B	B	B	C	B	B	B	E
D	C	D	C	D	D	C	C	C	B	C	D
E	B	E	E	E	B	E	E	D	D	D	D
F	B	F	E	F	D	F	F	F	F	E	E

Déterminer à l'aide de ces tables le chemin emprunté par un paquet de données envoyé du routeur A vers le routeur F.

2. On veut représenter schématiquement le réseau de routeurs à partir des tables de routage. Recopier sur la copie le schéma ci-dessous :



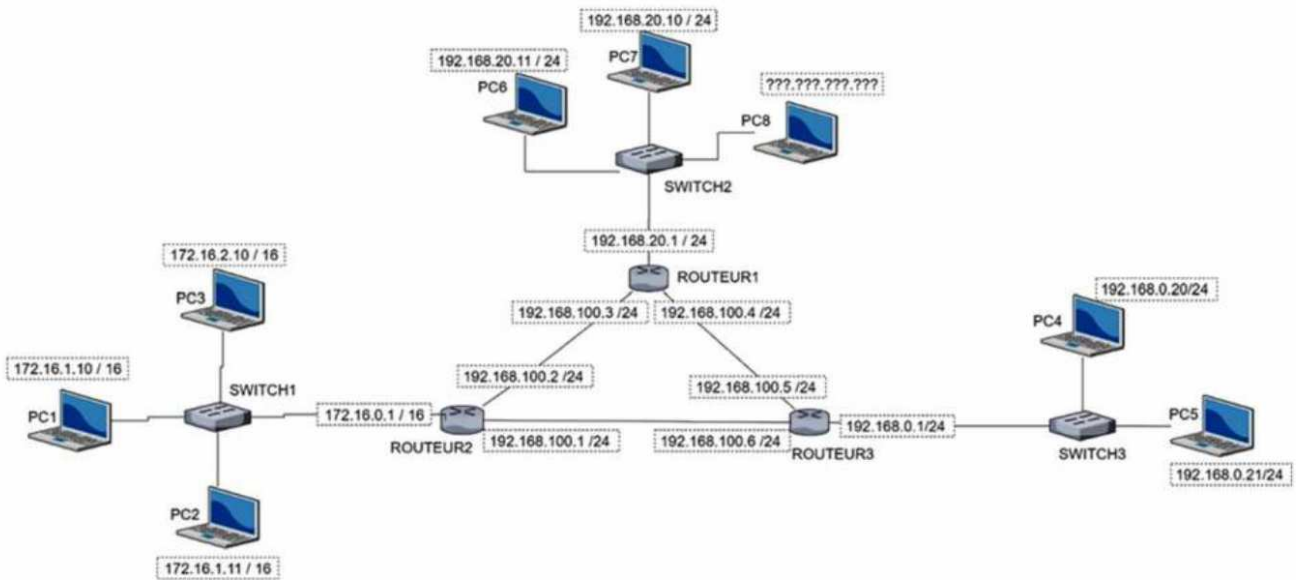
En s'appuyant sur les tables de routage, tracer les liaisons entre les routeurs.

4.4.5.2. SUJET 2 , EXERCICE 4**Etude d'un réseau informatique**

Soit un réseau informatique dont le schéma structurel simplifié est représenté ci-dessous. Il est composé de 8 PC, 3 switches, et 3 routeurs.

Dans cet exercice, on utilisera l'adressage CIDR composé d'une adresse IPv4 et d'une indication sur le masque de sous réseau. Par exemple : **172.16.1.10 / 16** signifie :

- Adresse IP : 172.16.1.10
- Masque de sous - réseau en notation CIDR : 16

**Partie A : ETUDE DE L'ADRESSAGE IP**

1. Sur le schéma ci-dessus, encadrer tous les sous - réseaux présents dans le réseau global sur le document réponse.

2. Etude du PC7 dont l'adresse IP est : 192.168.20.10 / 24.

2.a. Combien d'octets sont nécessaires pour composer une adresse IP(V4) ?

2.b. Compléter la ligne 2 du tableau du document réponse en convertissant la notation décimale de l'adresse IP en notation binaire.

La notation CIDR /16 pour une adresse IP signifie que le masque de sous - réseau a les 16 bits de poids fort de son adresse IP à la valeur 1. C'est-à-dire: 11111111.11111111.00000000.00000000.

2.c. Compléter la ligne 3 du tableau de l'annexe 3 en donnant le codage binaire du masque de sous - réseau en notation CIDR /24.

2.d. En déduire, à la ligne 4 du tableau de l'annexe, l'écriture décimale pointée du masque de sous - réseau.

L'adresse du réseau peut s'obtenir en réalisant un ET logique bit à bit entre l'adresse IP du PC7 et le masque de sous - réseau.

- Compléter la ligne 5 du tableau de l'annexe 2 avec l'adresse binaire du réseau.
- Compléter la ligne 6 du tableau avec l'adresse décimale du réseau.

3. Connexion du PC8 au réseau

Répondre au questionnaire sur le document réponse joint en cochant la ou les bonnes réponses.

Partie B : Une fonction pour convertir une adresse IP en décimal pointé en notation binaire.

On dispose de la fonction `dec_bin` :

- qui prend en paramètre d'entrée un nombre entier compris entre 0 et 255
- qui retourne une **liste** de 8 éléments correspondant à la conversion du nombre en écriture décimale en notation binaire. Chaque élément de cette liste est de type entier.

Exemples d'exécution de la fonction `dec_bin` :

- `dec_bin(10)` retourne la liste `[0, 0, 0, 0, 1, 0, 1, 0]`
- `dec_bin(255)` retourne la liste `[1, 1, 1, 1, 1, 1, 1, 1]`

Ecrire une fonction en langage Python que l'on appellera `IP_bin` qui:

- **prend** en paramètre d'entrée une liste de 4 entiers compris entre 0 et 255 correspondant à l'adresse IP en notation décimale
- **retourne** une liste de 4 listes correspondant à l'adresse IP en notation binaire.

La **fonction** `IP_bin` peut faire appel à la fonction `dec_bin`.

Exemple d'exécution de la fonction `IP_bin` :

```
>>> IP_bin([192, 168, 0, 1])
>>> [1, 1, 0, 0, 0, 0, 0, 0], [1, 0, 1, 0, 1, 0, 0, 0],
      [0, 0, 0, 0, 0, 0, 0, 0], [0, 0, 0, 0, 0, 0, 0, 1]]
```

Document réponse 2 (exercice 4)

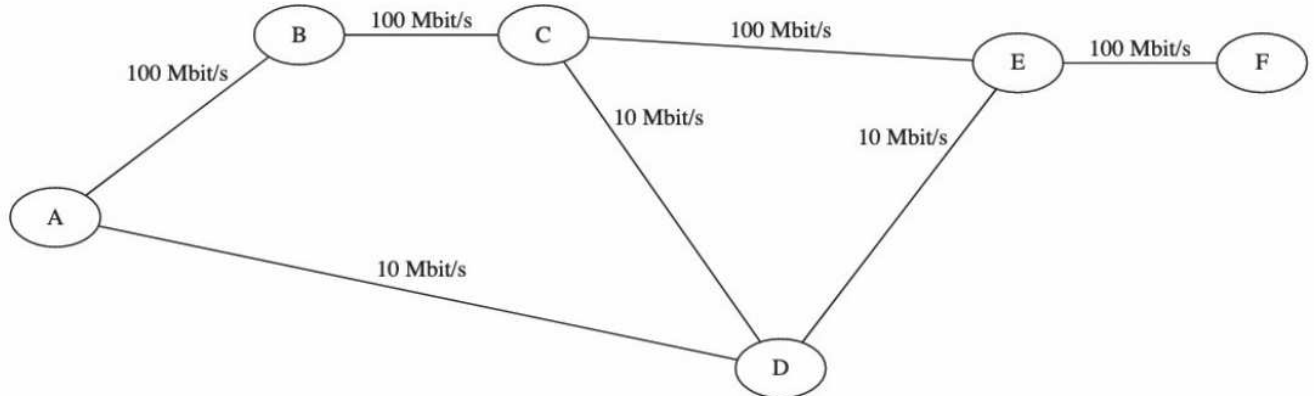
Adresse IP (V4) du PC7	Ligne 1	192	168	20	10
	Ligne 2	1 1 0 0 0 0 0 0		0 0 0 1 0 1 0 0	
Masque de sous réseau	Ligne 3				
	Ligne 4	255			
Pour obtenir l'adresse réseau binaire, on réalise un ET(&) logique entre chaque bit de l'adresse IP (ligne 2) et du masque de sous réseau (ligne3)					
Adresse du réseau	Ligne 5				
	Ligne 6		168		

On désire connecter le PC8 au réseau précédent. Parmi les propositions suivantes, cochez les adresses IP possibles pour le PC8 :

- ☐ 192.168.20.0
- ☐ 192.256.20.11
- ☐ 192.168.20.30
- ☐ 192.168.20.230
- ☐ 192.168.20.260
- ☐ 192.168.27.11

4.4.5.3. SUJET 3, EXERCICE 3**Partie B : Routage**

On considère le réseau modélisé par le schéma ci-dessous. Les routeurs sont identifiés par les lettres de A à F ; les débits des liaisons entre les routeurs sont indiqués sur le schéma.



1. Dans cette question, tous les routeurs utilisent le protocole RIP (distance en nombre de sauts).

On s'intéresse aux routes utilisées pour rejoindre F une fois les tables stabilisées.

Recopier et compléter sur la copie la table suivante :

Machine	Prochain saut	Distance
A		
B		
C		
D		
E		

2. Dans cette question tous les routeurs utilisent le protocole OSPF (distance en coût des routes). Le coût d'une liaison est modélisé par la formule

$$\frac{10^8}{d}$$

où d est le débit de cette liaison exprimé en bit par seconde.

On s'intéresse aux routes utilisées pour rejoindre F une fois les tables stabilisées.

Recopier et compléter sur la copie la table suivante :

Machine	Prochain saut	Distance
A		
B		
C		
D		
E		

3. Des protocoles RIP et OSPF, lequel fournit le routage entre A et F le plus performant en terme de débit ? Justifier la réponse.

- 1.** Expliquer pourquoi le protocole TCP-IP prévoit un découpage en paquets et une encapsulation des fichiers transférés d'un ordinateur à un autre via Internet.

- 2.** On souhaite modéliser un réseau informatique par un graphe pondéré pour identifier le chemin optimal pour un paquet.
 - 2.a.** Préciser ce que représentent les sommets et les arêtes du graphe.

 - 2.b.** Préciser si le protocole RIP utilise le nombre de sauts ou le délai de réception comme poids des arêtes.