



UE GLIN 102

Concepts de base en informatique

Année 2011 - 2012



Présentation générale

- *But*

Découvrir les concepts de base ..

Devenir un utilisateur averti des salles Rez-Ufr

- *Organisation*

Cours

Séances pratiques (TD - TP)

Evaluation



Salle informatique de Rez-UFR

- Rez-UFR

Dispositif complexe de **mutualisation** de ressources et de services

- Partie physique

Postes de travail, Serveurs,
Périphériques, Connectique ...

- Partie logicielle

Systèmes d'exploitation
Applications

- Fichiers



Connexions

- Sur toute machine du réseau
- Connexion : possible après la procédure d'inscription
- vous devez saisir :
 - votre Login
(appelé aussi nom d'utilisateur, identifiant ou pseudo)
on a standardisé :
1e lettre prénom. 7 premières lettres du nom
 - votre mot de passe (= Password)
Construit par vous (attention sécurité)



Environnement

- Un répertoire personnel (appelé Home) dans lequel vous pourrez créer des sous-répertoires de façon arborescente (ou hiérarchique) vous pourrez loger vos fichiers dans cet espace. gérer cet espace demande méthode et réflexion.
- Un espace de stockage : limité il faudra éviter la saturation (grosses impressions, téléchargement d'images ...)
- Un ensemble de logiciels accessibles



Contenu

- Généralités et Architecture
- Système d'exploitation
- Logiciels de base ... développement
- Web et Réseaux



Contenu

- **Généralités et Architecture**
- Système d'exploitation
- Logiciels de base ... développement
- Web et Réseaux

Généralités

- Comment fonctionne un ordinateur ?

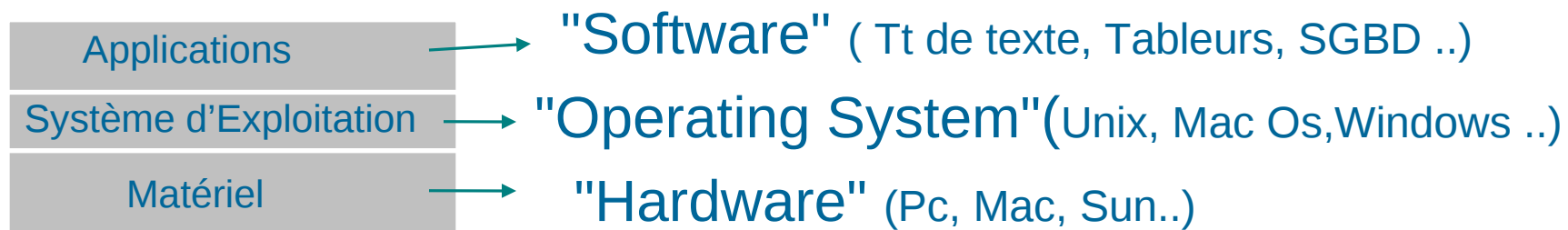
on peut distinguer 3 couches :

- Applications

un utilisateur interagit surtout avec cette couche

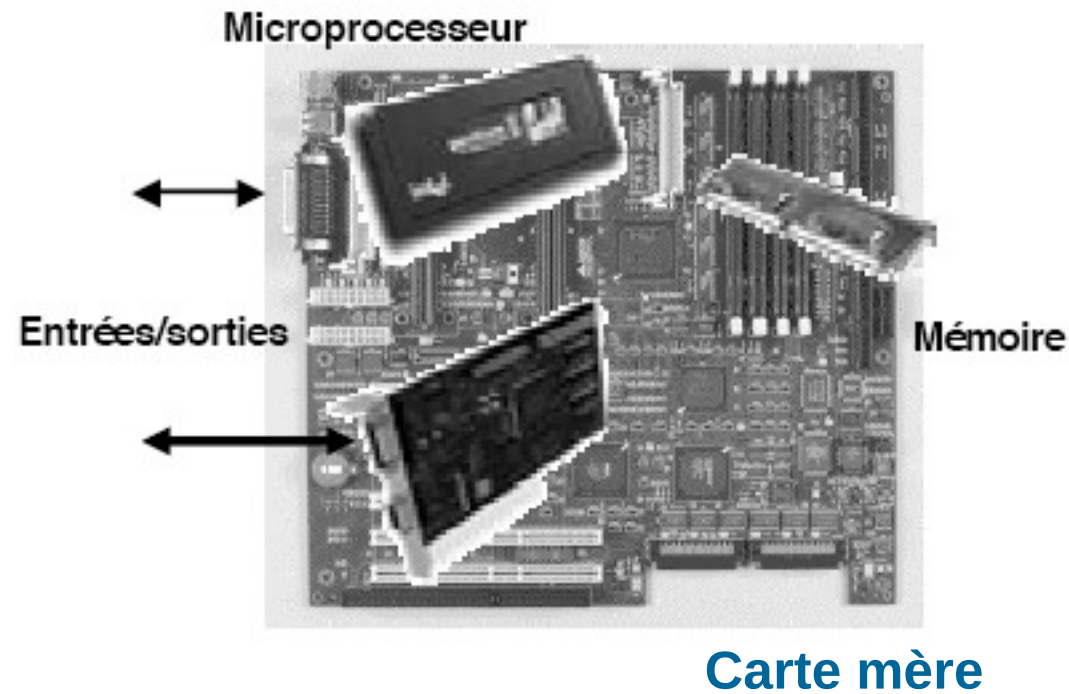
- Système d'Exploitation

- Matériel



Architecture matérielle

Aspect physique (voir architecture)





Information en machine

bit et octet

En machine, l'unité est le **bit** (**b**inary **dig**it)
valant 0 ou 1

En mémoire, l'unité est un "paquet" de 8 bits : l'**octet**



Information en machine

codage

un octet ou un groupe d'octets
peuvent représenter plusieurs choses

un caractère (imprimable),
un nombre entier,
un nombre flottant (notation scientifique),
une image ...



Codages

- exemple 1: nombre entier (sur 2 octets)
en base 2 (binaire)

Exemple

0	0	0	1	1	0	1	1	0	0	0	0	1	1	0	1
2^{15}	2^{14}	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

le nombre 1101100001101 en base 2 vaut

$$2^{12} + 2^{11} + 2^9 + 2^8 + 2^3 + 2^2 + 2^0 = 4096 + 2048 + 512 + 256 + 8 + 4 + 1 = 6925$$

pour le signe - c'est plus complexe, mais

Sur 2 octets on représentera les entiers entre -32768 et + 32767

l'utilisateur n'aura pas besoin de compter en binaire

Codages

- Exemple 2 : les nombres flottants (sur 8 octets)



C'est presque la notation scientifique mais en binaire

Signe (s) : codé sur 1 bit, 0 pour positif, 1 pour négatif

Partie fractionnaire (pf) : entier (positif) sur 47 bits dont le 1er chiffre est non nul (normalisé)

Exposant (e) entier signé sur 16 bits

correspondant à la puissance de 2 (ou 16 ...)

le nombre correspondant à $\langle s, pf, e \rangle$ est $s \cdot 0, pf * 2^e$

Exemple : $\langle 1, 11, 1 \rangle = - (0,11 * 2^1)_2 = - (0,75 * 2)_{10} = -1,5$



Architecture (C. Boksenbaum)

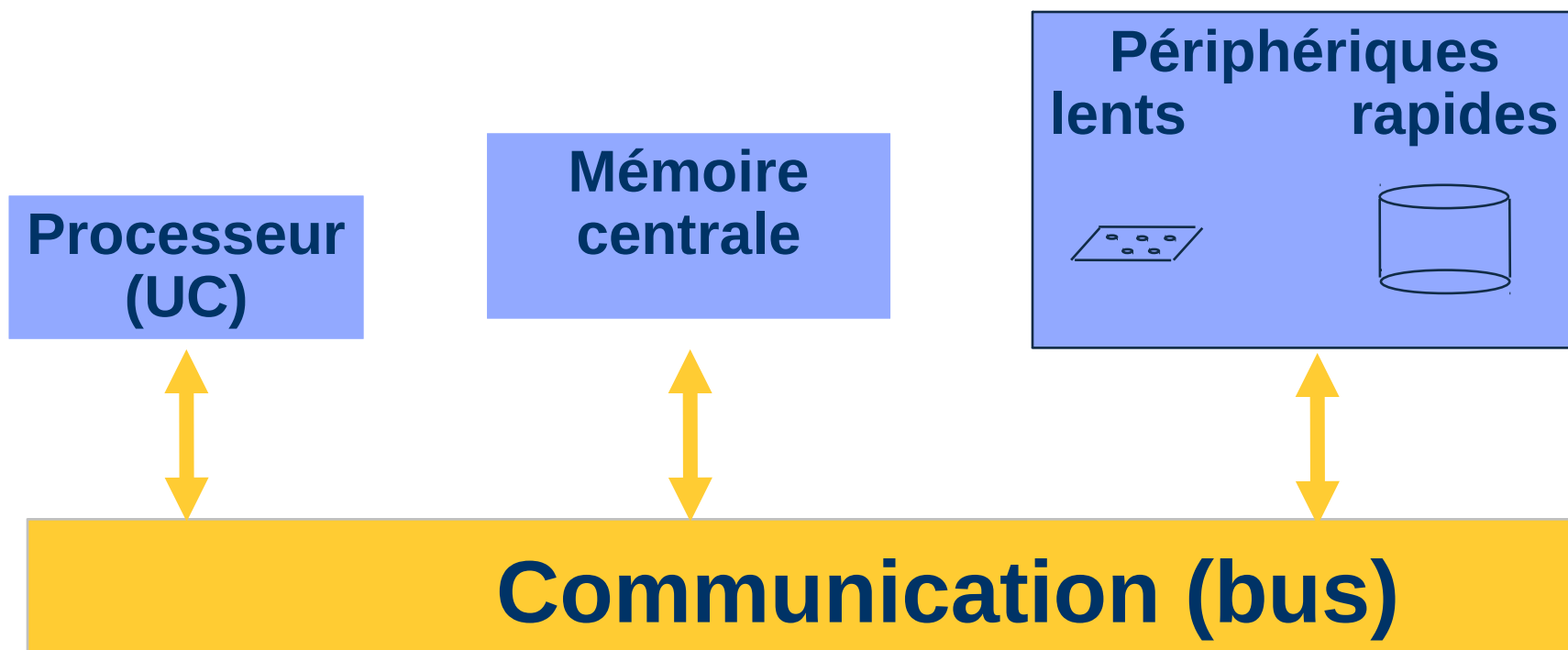
- Schéma général d'un ordinateur
- Quelques chiffres ... pour comprendre
- Quelques termes
- Fonctionnement d'une UC

Architecture



- Schéma général d'un ordinateur
- Quelques chiffres ... pour comprendre
- Quelques termes
- Fonctionnement d'une UC

Schéma général d'un ordinateur



Attention : Il y a des variantes

Schéma général d'un ordinateur

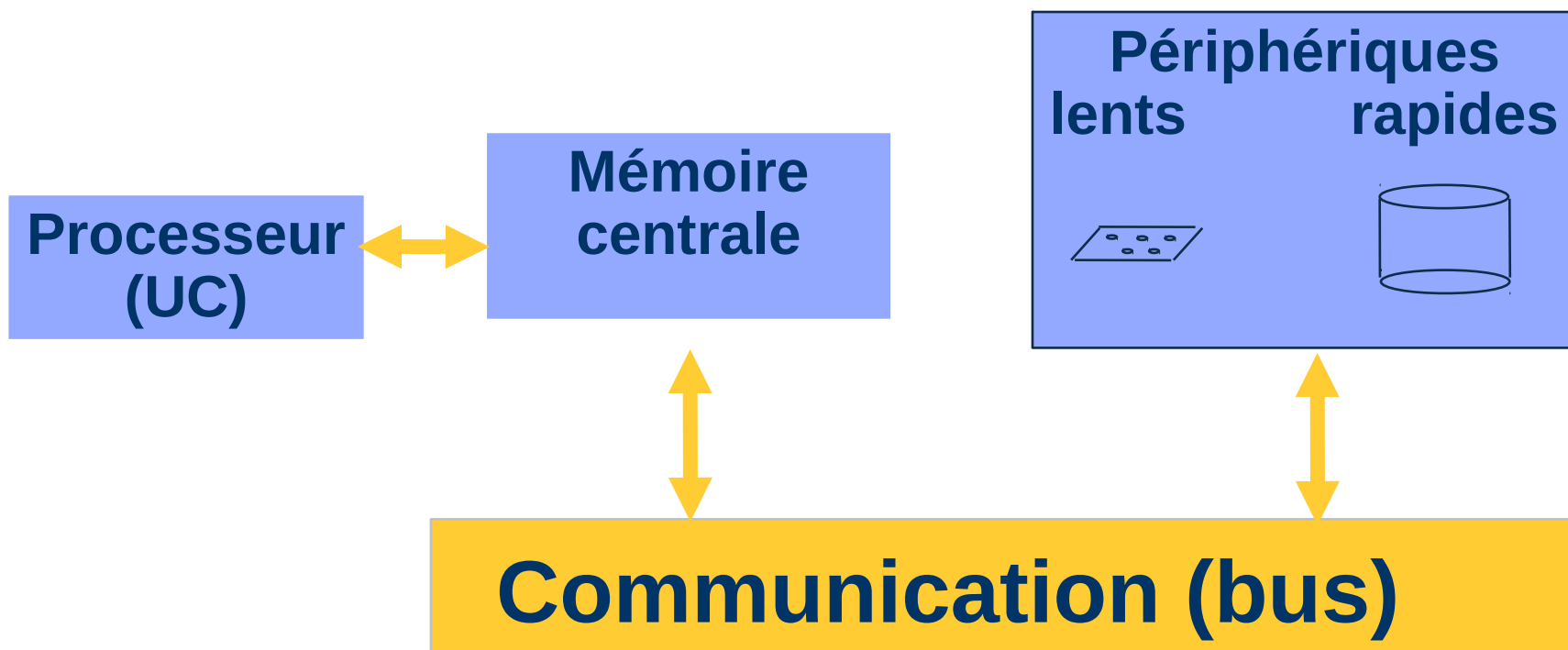


Schéma général d'un ordinateur

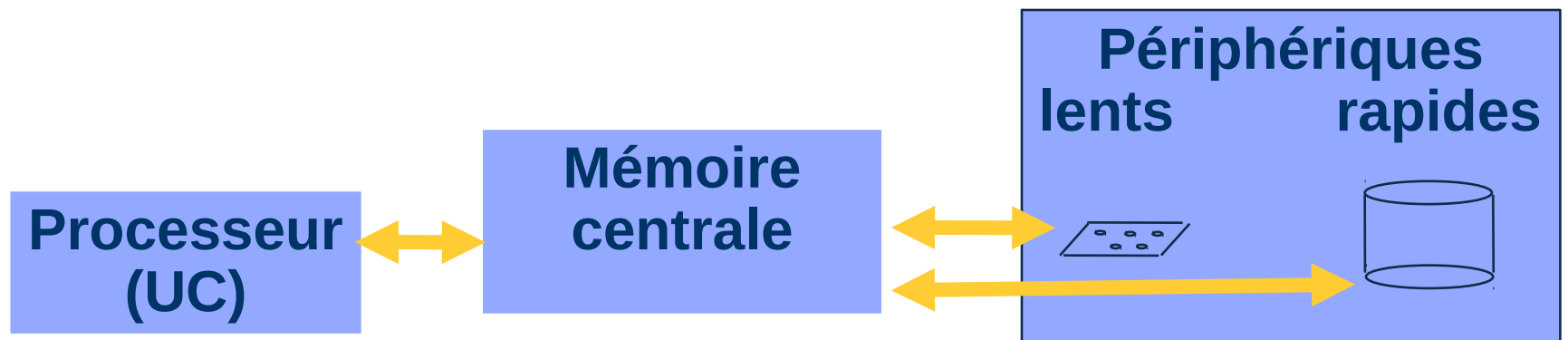




Schéma général d'un ordinateur

Il y a **2 sortes d'activités** dans un ordinateur :

- **exécution des instructions** par l'UC
- **échange d'information** entre mémoire centrale et périphériques

Schéma général d'un ordinateur

L'UC (unité centrale) exécute des instructions logées en **mémoire centrale**

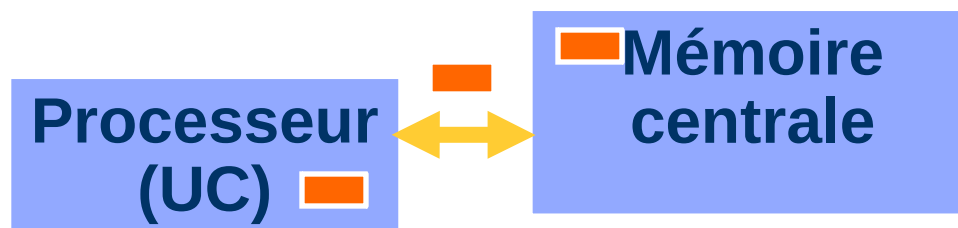


Schéma général d'un ordinateur

L'échange d'information a lieu
entre la mémoire centrale et les périphériques
(entrées-sorties)

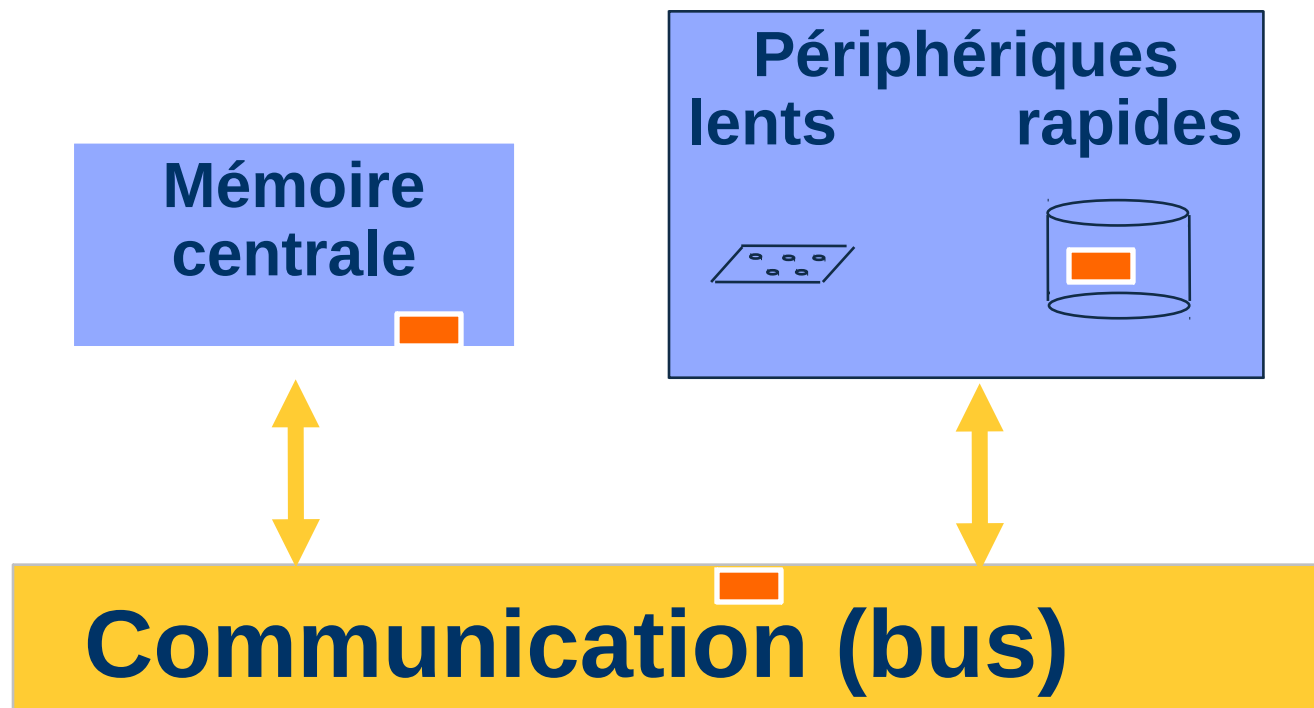


Schéma général d'un ordinateur

L'échange d'information a lieu
entre la mémoire centrale et les périphériques
(entrées-sorties)

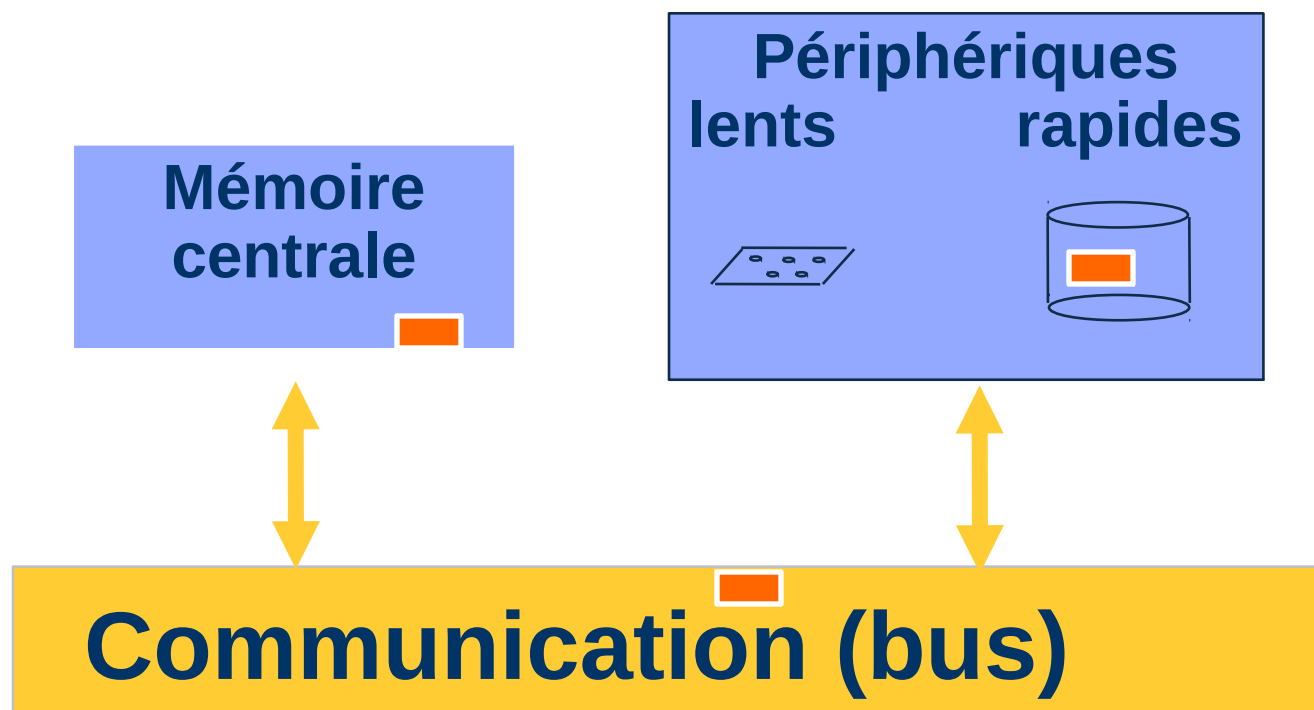




Schéma général d'un ordinateur

Entrées-Sorties (du point de vue de la mémoire centrale)

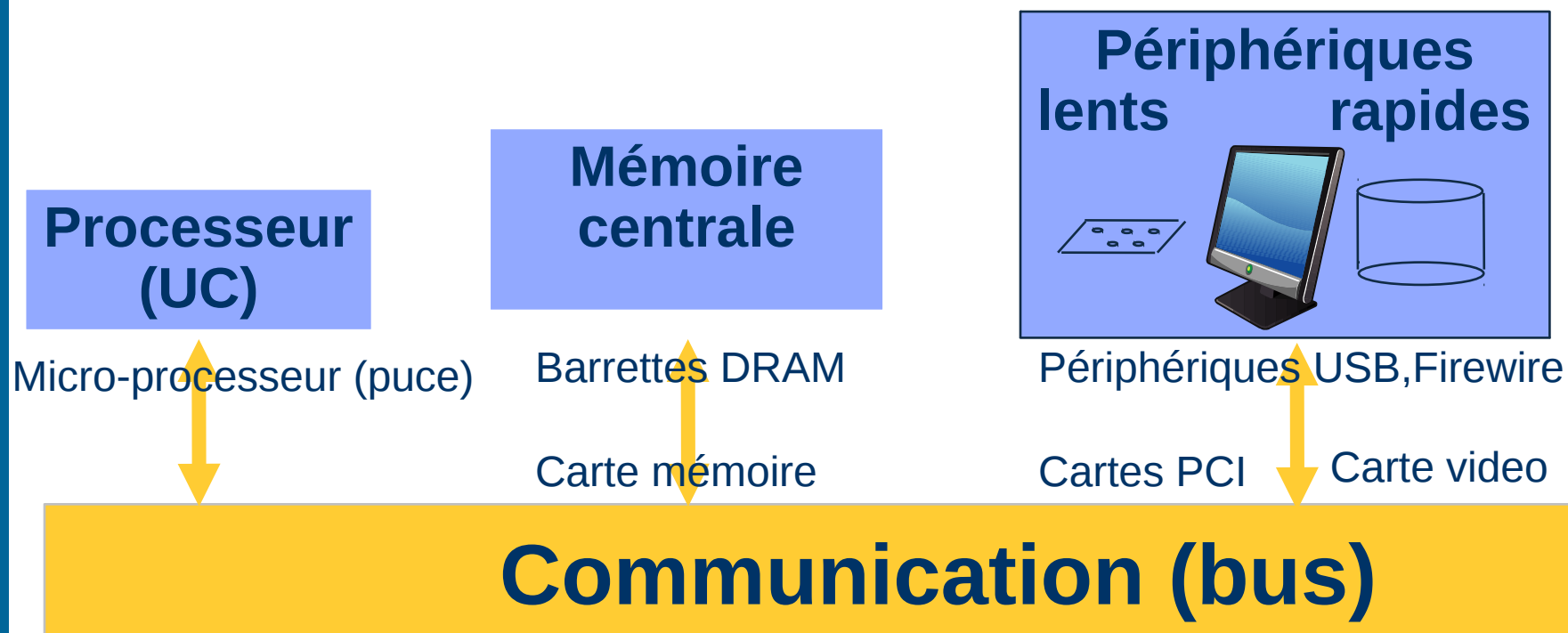
elles sont démarrées par une **instruction de l'UC**

la taille du bloc échangé peut varier :

- 1bit (transmission série (à 56 Kb/s par exemple))
- un octet (clavier)
- un secteur (disque) de 512 octets à 4 kO

Schéma général d'un ordinateur

Aspect physique



Architecture

- Schéma général d'un ordinateur
- ➔ ● Quelques chiffres ... pour comprendre
- Quelques termes
- Fonctionnement d'une UC



Quelques chiffres ... pour comprendre

pour compter vite :

en informatique on utilise des puissances de 2
(on verra pourquoi plus tard)

l'approximation

$$1024 = \boxed{2^{10} \# 10^3} = 1000$$

permet de calculer vite.

ex. combien peut-on coder de couleurs sur 32 bits ?

réponse : $2^2 \cdot 2^{30} \# 4 \cdot 10^9$



Quelques chiffres ... pour comprendre

Attention aux échelles de grandeur !!

capacité (d'octets) :

1	10^3	10^6	10^9	10^{12}
	kilo (ko)	méga (Mo)	giga (Go)	tera (To)

période :

s	ms (= 10^{-3} s)	μ s (= 10^{-6} s)	ns (= 10^{-9} s)
---	--------------------	-------------------------	--------------------

fréquence correspondante

1 Hz	KHz (= 10^3 Hz)	MHz (= 10^6 Hz)	GHz (= 10^9 Hz)
------	-------------------	-------------------	-------------------



Quelques chiffres ... pour comprendre

exemple d'utilisation :

un P4 / 3 GHz sdram 256 M0 DD 80/ CDrw 52x/
reseau 10/100 wifi

est l'abréviation du modèle où

- l'horloge du processeur Pentium 4 bat à 3 Ghz
- la mémoire centrale contient 256 MégaOctets
- le disque dur contient 80 GigaOctets
- le lecteur de CD-rom est aussi graveur (w)
à vitesse 52 (/ audio normal)
- la carte réseau fonctionne à 10 ou 100 Mbits/s
avec transmission sans fil vers un routeur



Quelques chiffres ... pour comprendre

Les échelles de temps

il y a environ 100 000s en 1 jour.

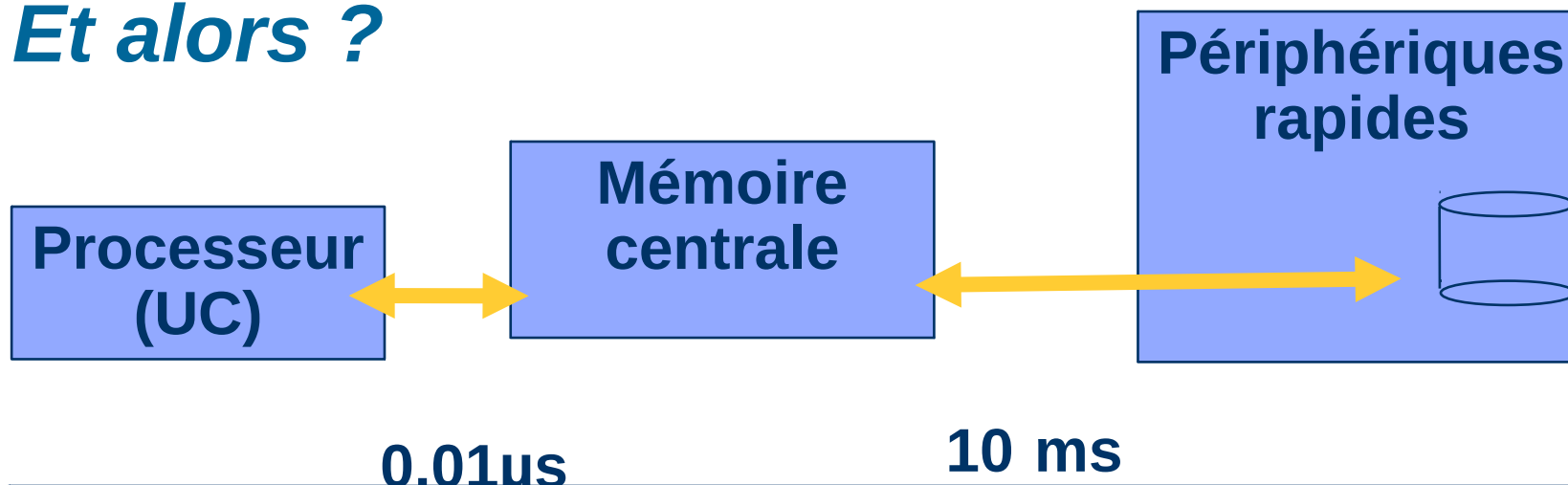
3 GHz : c'est la fréquence d'horloge du processeur
il est capable de faire quelque chose en 1/ 3 000 000 000s
une instruction utile se fera plutôt en **0,01 μ s**
(1 /100 000 000s)

un transfert mémoire-disque dur se fait en **10 ms**
(soit 1 000 000 instructions)

Et alors ?

Quelques chiffres ... pour comprendre

Et alors ?



Si le processeur attend une entrée-sortie, pour lui, c'est comme attendre 10 jours ...

Le matériel et le logiciel seront organisés pour éviter cette attente ...



Quelques chiffres ... pour comprendre

Les échelles de temps

une image 1000 x 1000 a 10^6 points.

en 2 couleurs (noir et blanc) il faut 10^6 bits pour la représenter.

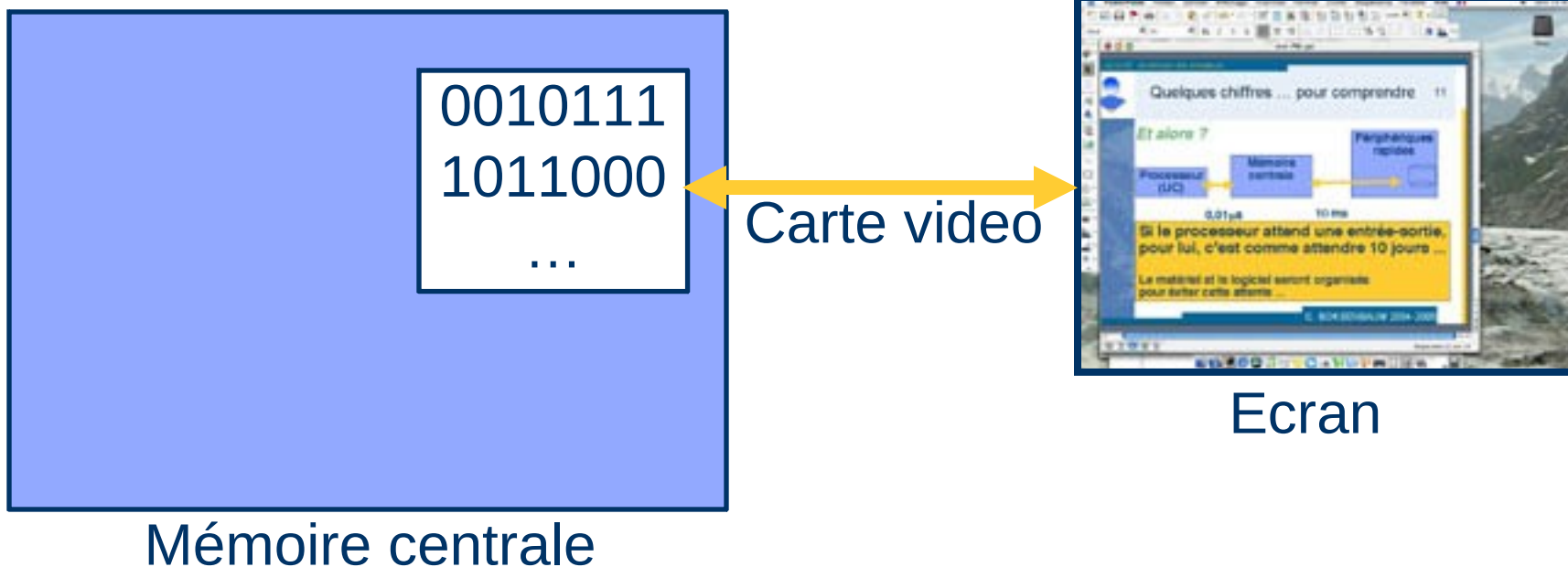
en couleur (sur 32 bits) il faut $32 \cdot 10^6$ bits pour la représenter.

Envoyer une image, à 10 Mb/s, prendrait 3 s :

- intérêt des techniques de compression (MPEG)
- affichage video \neq transfert à un périphérique
(affichage = écriture en mémoire centrale)

Quelques chiffres ... pour comprendre

- affichage video \neq transfert à un périphérique
(affichage = écriture en mémoire centrale)





Quelques chiffres ... pour comprendre

Les avancées trop rapides de la technologie (capacité) compliquent l'architecture :

1980: adressage physique de 1Mo maximum
(dont le dernier tiers est utilisé pour les périphériques)

1985: les PC atteignent cette limite.

1965 : espace virtuel (utopique) maximum de 2Go

2004: les gros PC atteignent cette limite.

Idem pour les disques :

1990: 1 Go maximum,

2004 : 80 Go en 2004 pour les PC courants

1 To pour les plus gros.

Comment gérer ces nouvelles capacités ?

Architecture

- Schéma général d'un ordinateur
- Quelques chiffres ... pour comprendre
- Quelques termes
- Fonctionnement d'une UC





Quelques termes

RAM (random access memory) = mémoire centrale sous forme de **barrettes** (128 Mo, 256 Mo...)

temps d'accès (en 2004) : 80 ns

mais le **cache** permet un temps moyen de 10 ns

La RAM est **volatile** : contenu perdu en cas de panne

Avec les grandes capacités actuelles des mémoires centrales (-> 2 Go), on peut y stocker :

- des grandes images,
- des bases de données importantes,
- des matrices numériques importantes



Quelques termes

ROM (read only memory) = mémoire en lecture seule sous forme de **puces** remplies en usine.

La ROM est **non-volatile**.

sert au **démarrage** (= bootstrap) des ordinateurs et aux **sous-programmes de base** (jusqu'à 256 Mo)



Quelques termes

CD-ROM (disque compact) = en lecture seule sous forme de **disques gravés** par laser (700 Mo). Les disques **audio** sont lisibles par ordinateur. Le temps d'accès est de l'ordre de 100 ms. La gravure (unique) est une **opération lente**.

DVD-ROM en lecture seule sous forme de **disques gravés** par laser (4,7 Go). Les disques **video** sont lisibles par ordinateur.

Sur les **CD et DVD ré-inscriptibles (rw)**, on peut écrire 1000 fois environ.



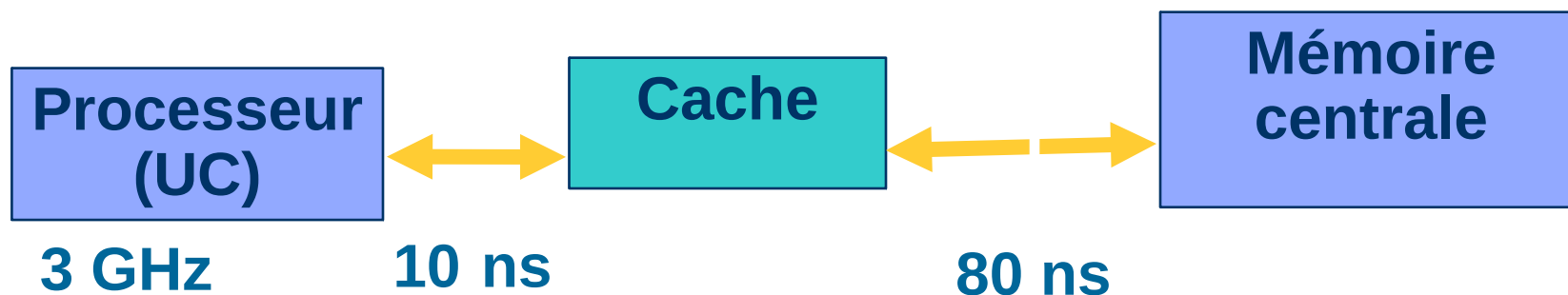
Quelques termes

Une clé **USB** est une mémoire électronique, considérée comme un **disque amovible** par le système.

Sa grande capacité (256 Mo, 1 Go ...) et son temps d'accès assez rapide en font une mémoire de sauvegarde idéale. **(non-volatile)**

Quelques termes

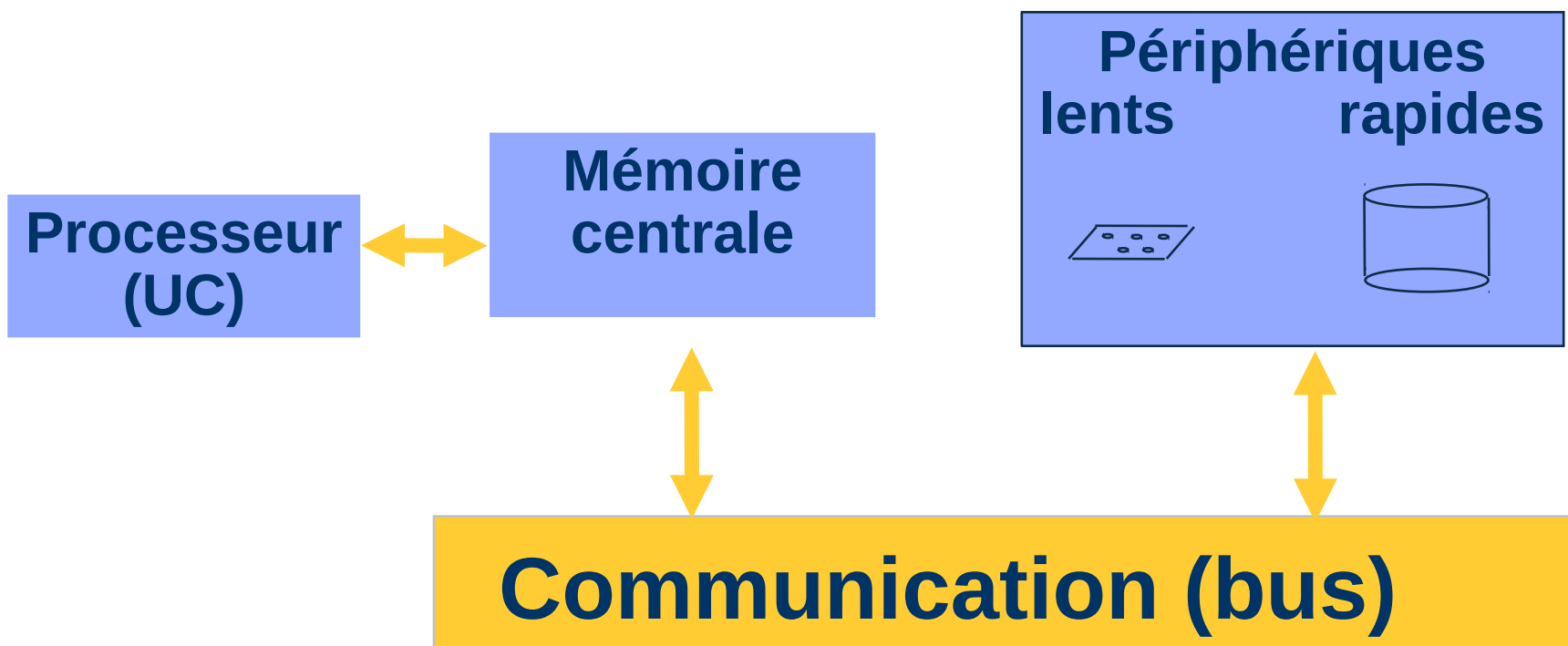
Cache : petite mémoire entre le processeur et la mémoire centrale (habituellement 100 ko - 1 Mo)



Avec le cache, le temps moyen d'accès est proche de 10 ns

Quelques termes

Bus : liaison commune à tous
facile de se brancher dessus (attention aux conflits !!)





Quelques termes

Jeu d'instructions (instruction set) : ensemble d'actions effectuées par le processeur

On distingue des jeux d'instructions

- de **même famille** (i8086, i8486, Pentium 4, Celeron, AMD ...)
- et ceux **incompatibles** (pentium et PowerPc)



Quelques termes

ASCII

(American Standard Code for Interchange of Information)
code international de caractères sur 8 bits

ne suffit pas pour les caractères spéciaux
de tous les pays.

Unicode

code international des caractères sur 16 bits

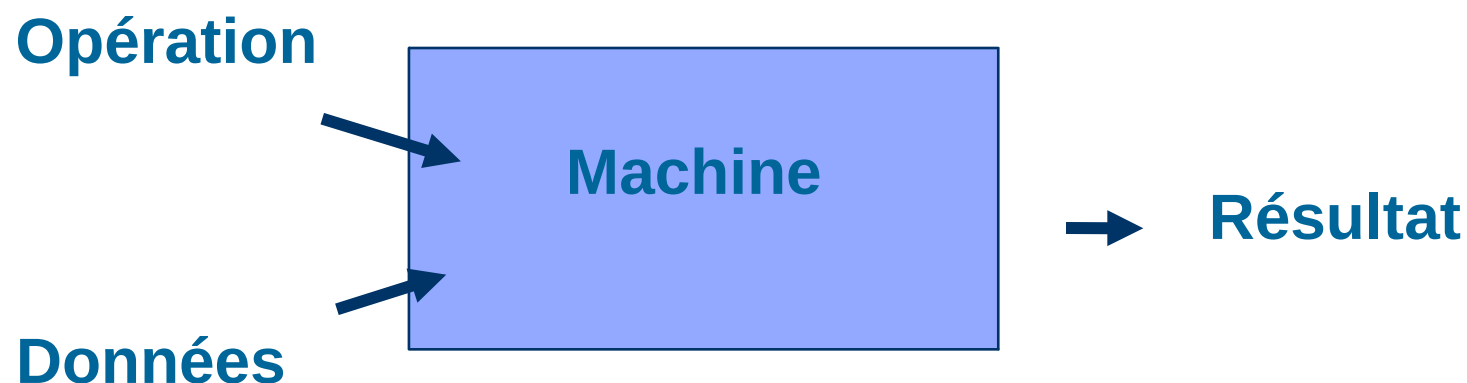
pour coder les caractères spéciaux
de tous les pays.

Architecture

- Schéma général d'un ordinateur
- Quelques chiffres ... pour comprendre
- Quelques termes
- ➔ ● Fonctionnement d'une UC

Fonctionnement d'une UC

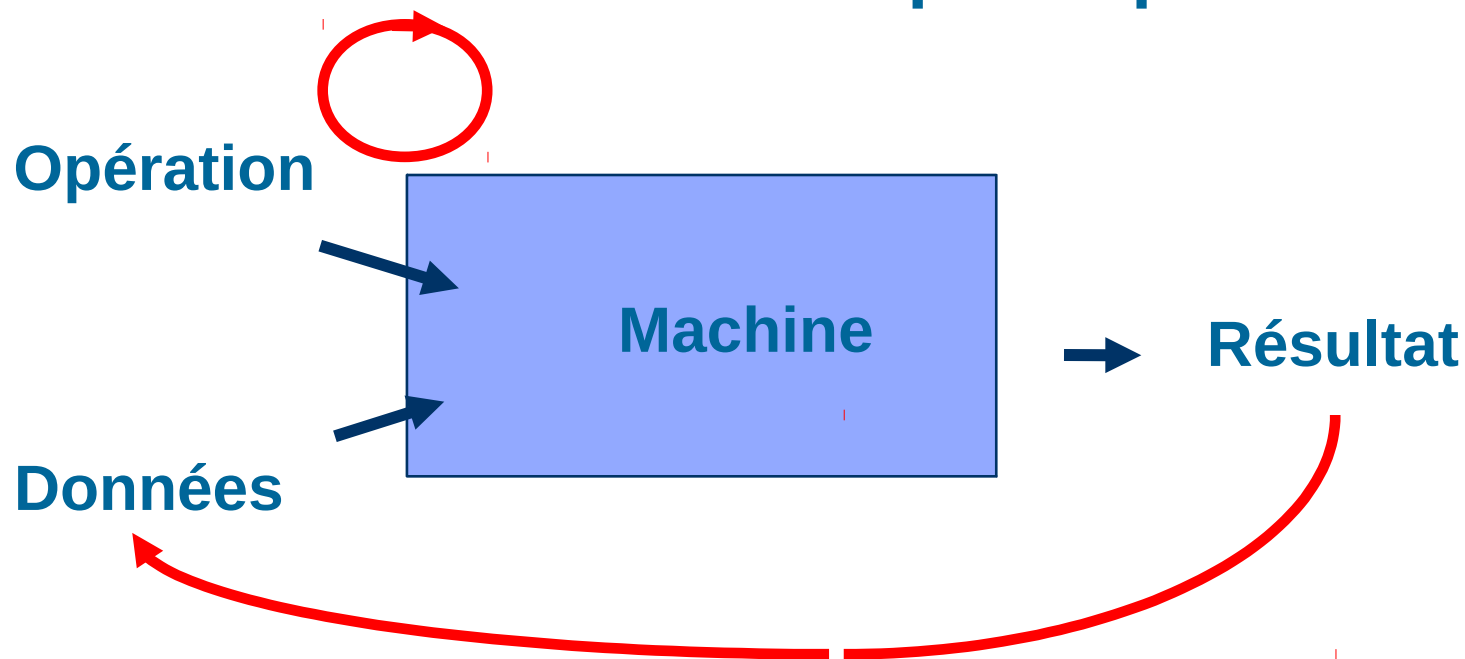
La calculette simple



**Toutes les informations
restent « stockées » en dehors de la machine**

Fonctionnement d'une UC

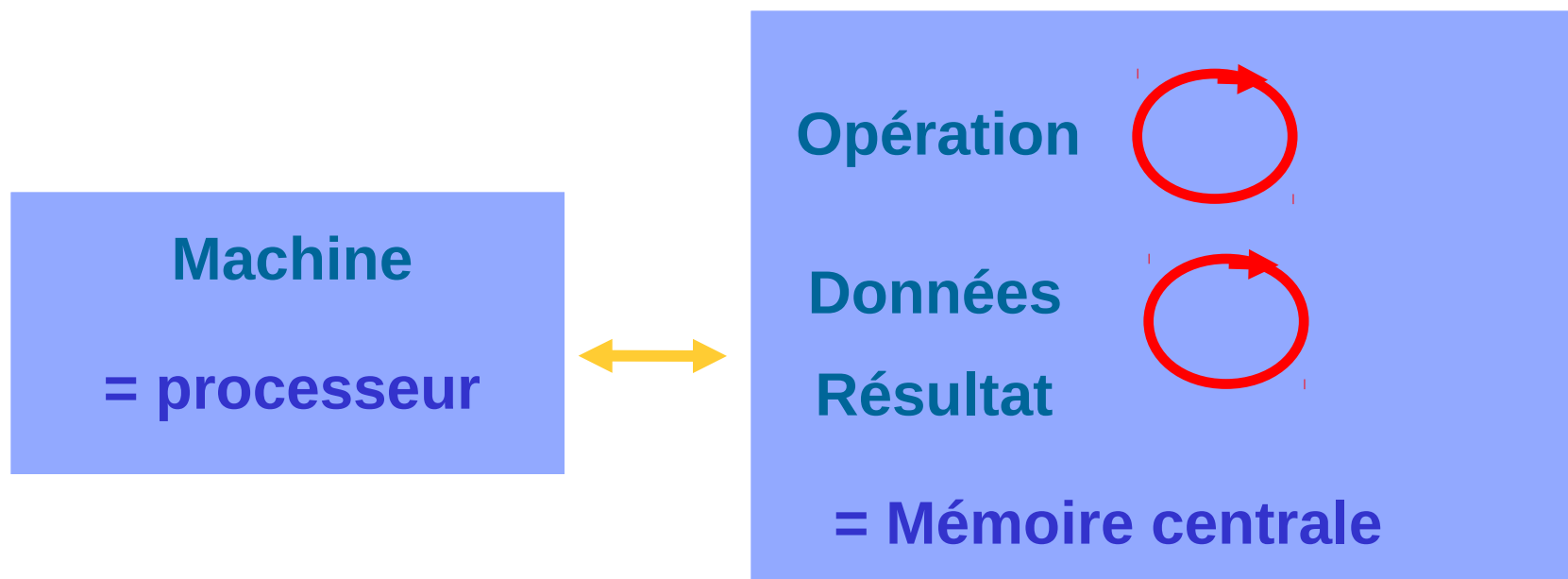
La calculette sophistiquée



les résultats "reviennent" en données
Les opérations s'enchaînent

Fonctionnement d'une UC

L'ordinateur





Fonctionnement d'une UC

Cycle de l'unité centrale

Répéter

chercher l'instruction à exécuter

chercher les données

exécuter l'instruction

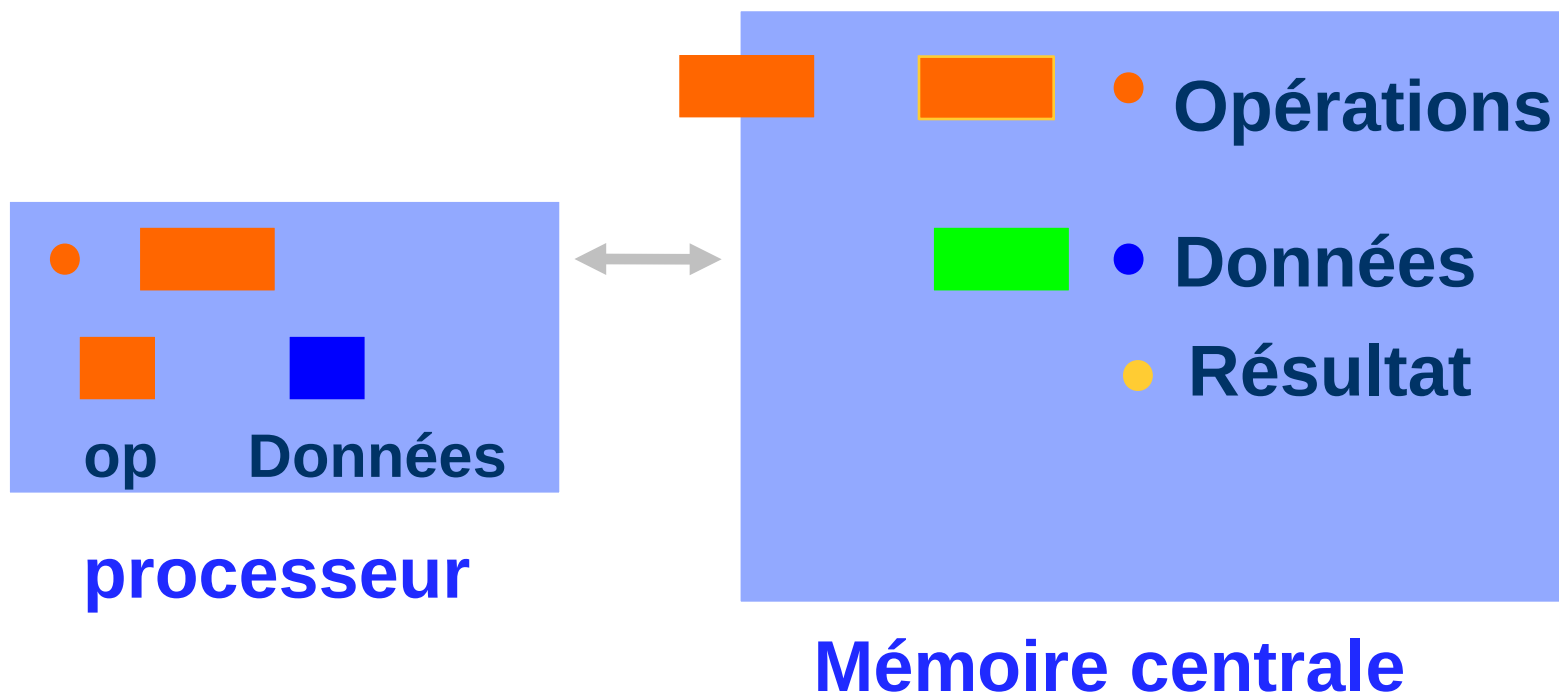
[ranger le résultat]

déterminer l'instruction suivante

sans fin

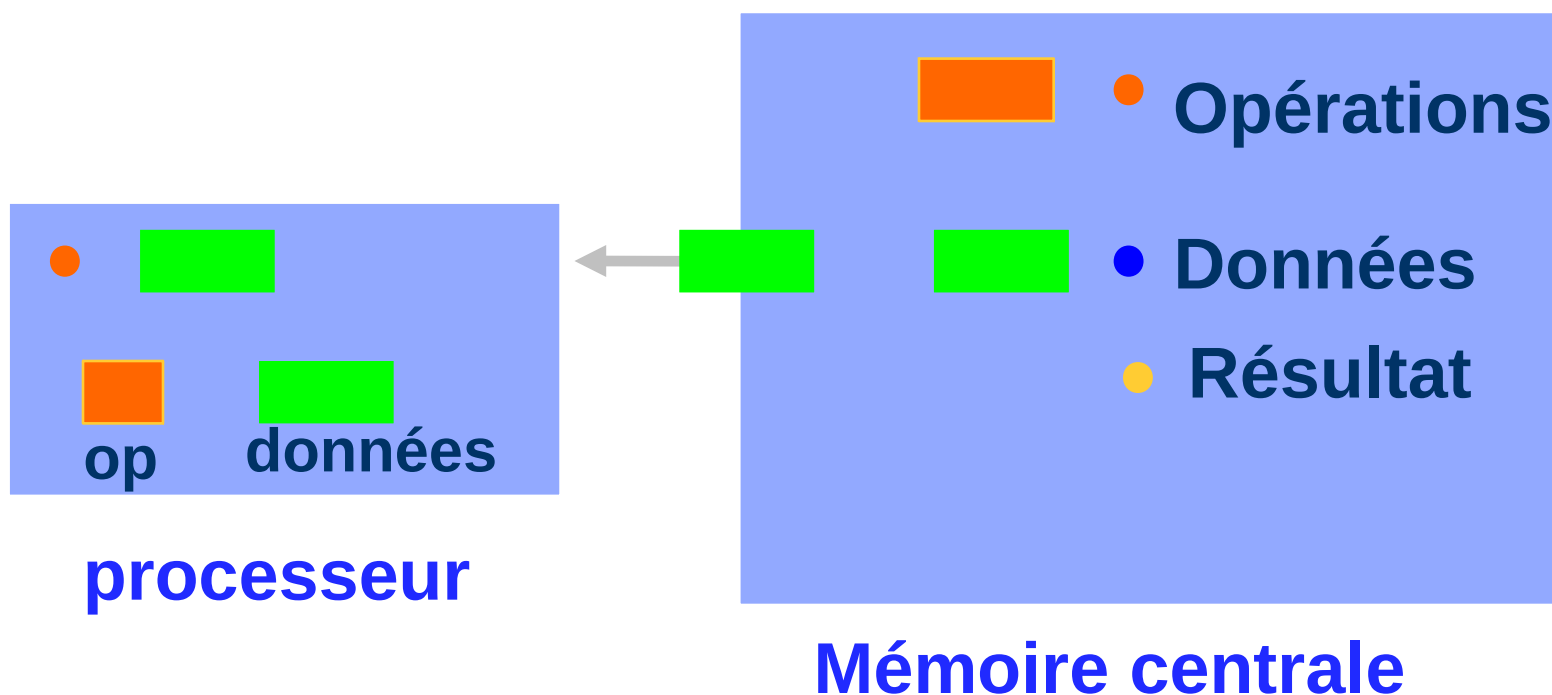
Fonctionnement d'une UC (animation)

Chercher l'instruction à exécuter



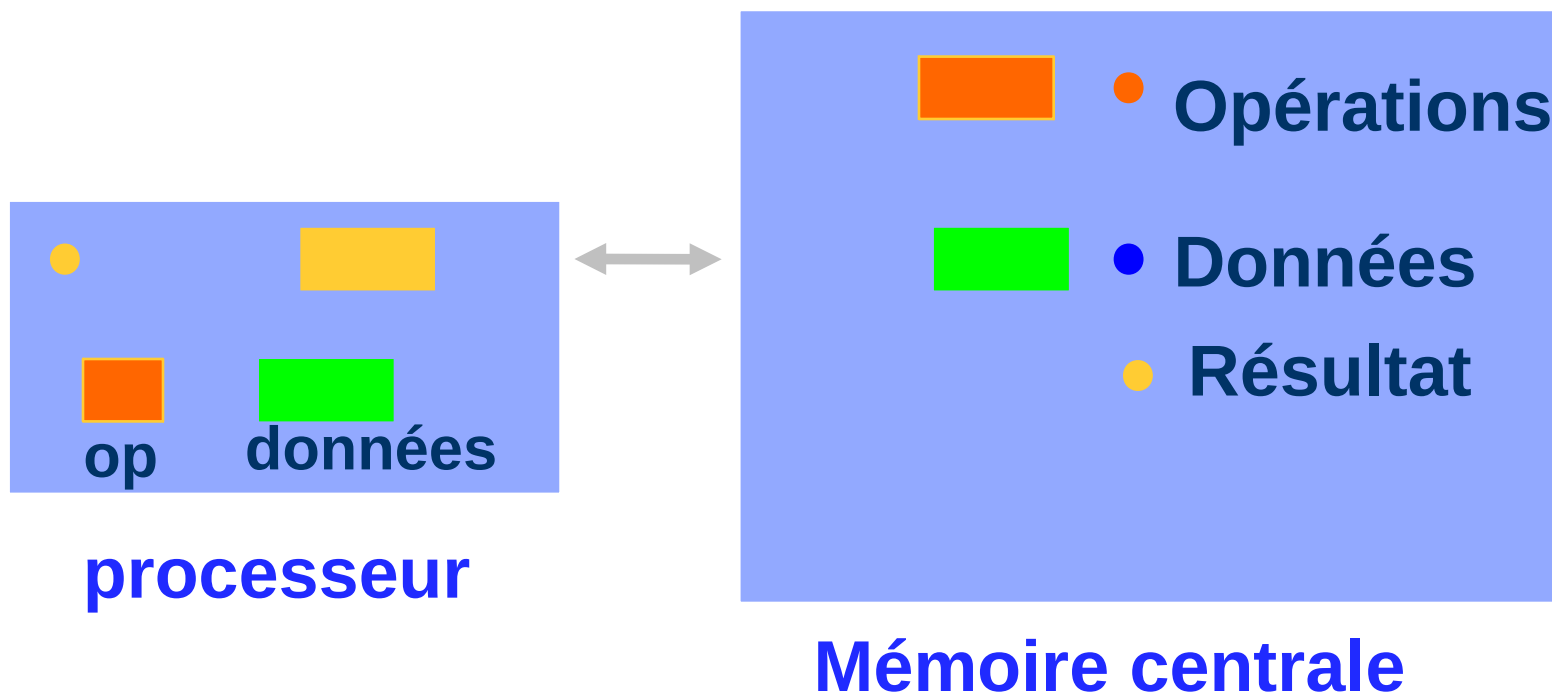
Fonctionnement d'une UC (animation)

Chercher les données



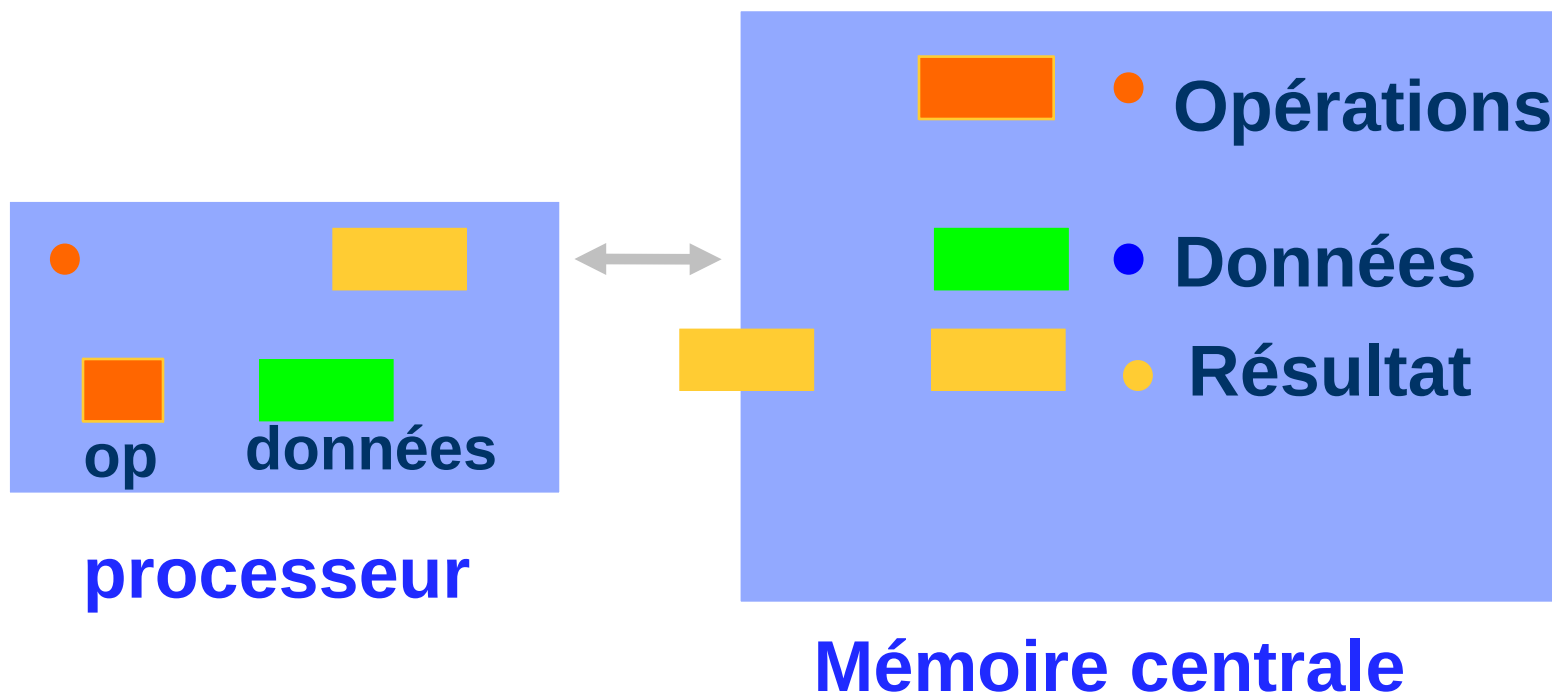
Fonctionnement d'une UC (animation)

Exécuter l'instruction



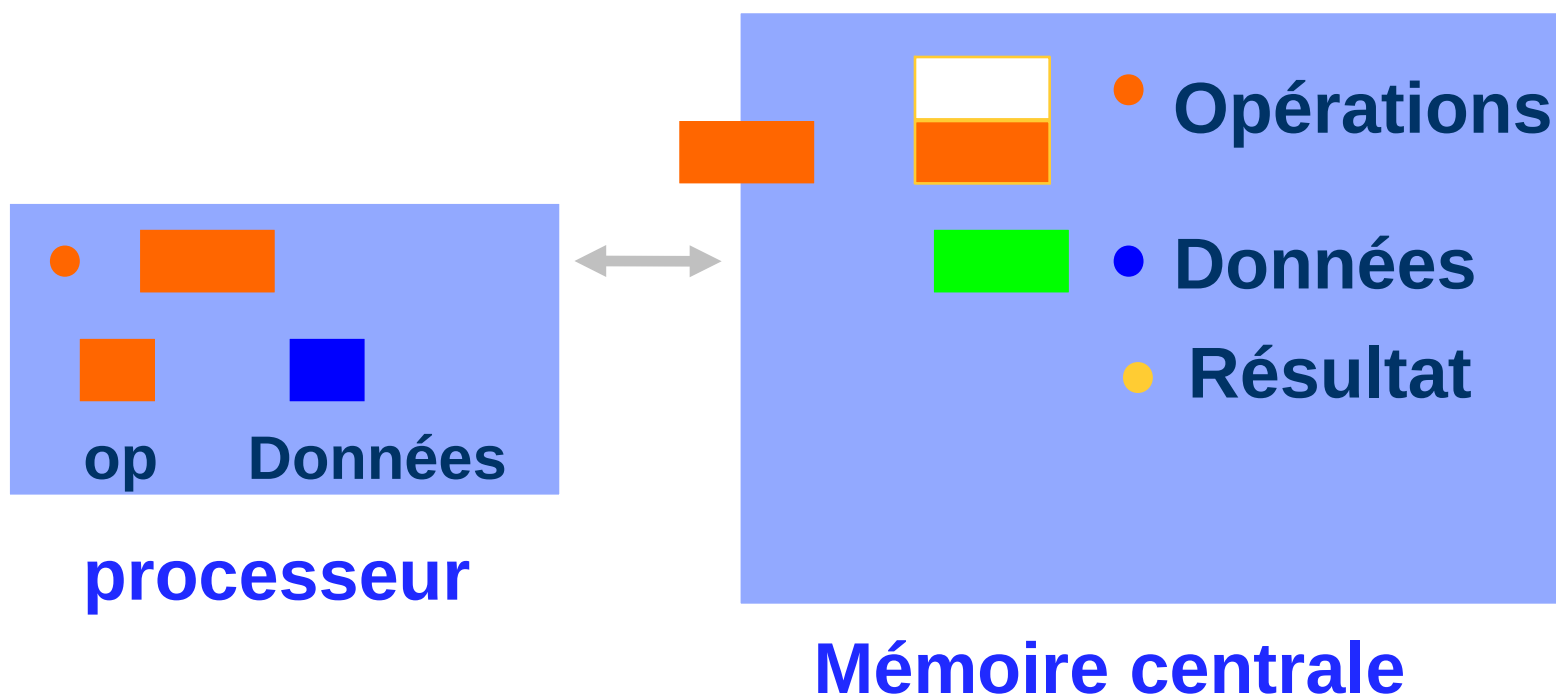
Fonctionnement d'une UC (animation)

Ranger le résultat



Fonctionnement d'une UC (animation)

... chercher l'instruction à exécuter





Fonctionnement d'une UC

Important

les instructions et les données

- logent **ensemble** dans la mémoire centrale
- sont codées de **façon stricte**



Contenu

- Généralités et Architecture
- **Systeme d'exploitation**
- Logiciels de base ... développement
- Web et Réseaux



Systemes d'exploitation (M. Leclere)

- Introduction : du matériel au logiciel
- Quelques concepts « système » vus du côté de l'utilisateur
- Rôles d'un système d'exploitation
- Les interpréteurs de commandes
- La gestion de fichiers
- La protection des données

Systemes d'exploitation

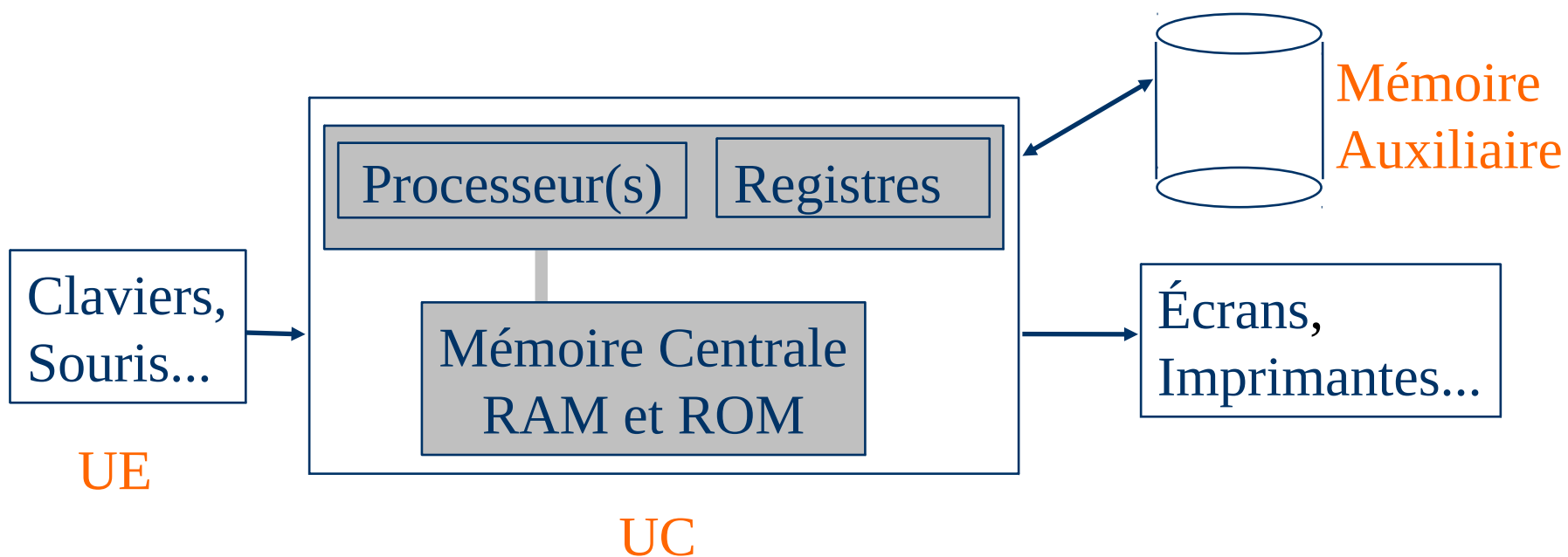


- Introduction : du matériel au logiciel
- Quelques concepts « système » vus du côté de l'utilisateur
- Rôles d'un système d'exploitation
- Les interpréteurs de commandes
- La gestion de fichiers
- La protection des données



Introduction : du matériel au logiciel

- Au cœur d'un système informatique, on trouve un ordinateur (**hardware**)





Fonctionnement de l'ordinateur

- Dès la mise sous tension, le processeur démarre **son cycle de travail**

Répéter

lire une instruction de la Mémoire Centrale ;
analyser l'instruction ;
exécuter l'instruction ;
sans fin



Le langage machine

- Pour que l'ordinateur réalise un travail, il faut que des instructions soient présentes en mémoire centrale (MC)
 - Cela nécessite le **chargement** d'un programme (une séquence d'instructions) d'une mémoire auxiliaire vers la MC
- Chaque processeur accepte un ensemble bien défini d'instructions qui constitue le **langage machine**
 - Ces langages sont rudimentaires et différents d'un processeur à l'autre (problèmes de compatibilité)

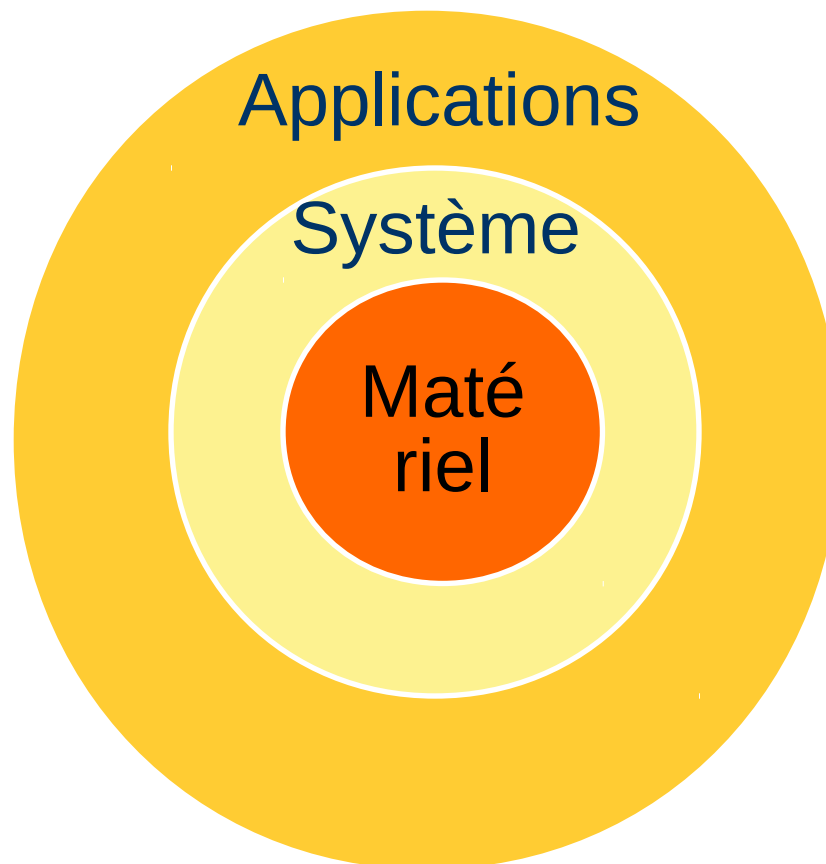


Les langages évolués

- Pour faciliter le travail de programmation, des **langages de haut niveau** ont été développés (Cobol, Fortran, LISP, C, Pascal, Ada, Smalltalk, Java...)
 - indépendants du matérieldonc assurant la portabilité des programmes
 - Avec des bibliothèques de fonctions pré-programmées
- Il faut alors disposer d'un **compilateur** qui traduit en programme exécutable (instructions en langage machine) le programme source (instructions en langage évolué)



Organisation d'un système informatique



Modèle en couches



Les programmes (**software**)

Le **système d'exploitation** est un ensemble de programmes

- qui "cache" le matériel aux programmes
(en gérant aussi ce matériel)
- qui fournit des services aux applications en terme de concepts non physiques (fichiers, processus)
- invocables par des **commandes** passées à un interpréteur



Les programmes (**software**)

Les applications sont des programmes :

- Adaptés à un service "pointu" (="métier")

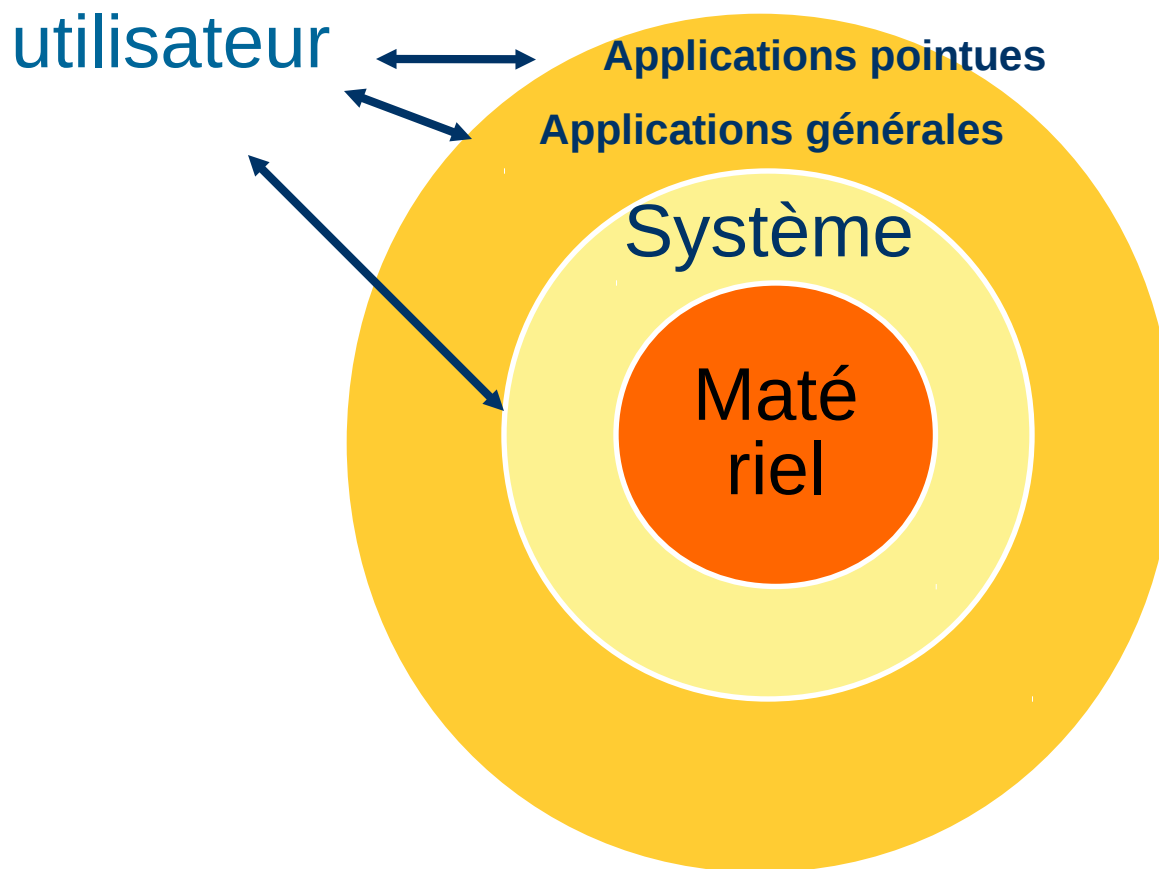
Ex : service bancaire, gestion de scolarité

- Généraux, utiles à (presque) tous les utilisateurs

Ex : traitement de texte, tableur, gestion de courrier, navigateur



Organisation d'un système informatique



Interactions de l'utilisateur



L'utilisateur

L'utilisateur courant interagit avec

- les services "pointus" (ou métier)
- les services généraux dans le travail de bureau
- Le système : pour gérer ses fichiers et ses dossiers (principalement sous forme graphique)



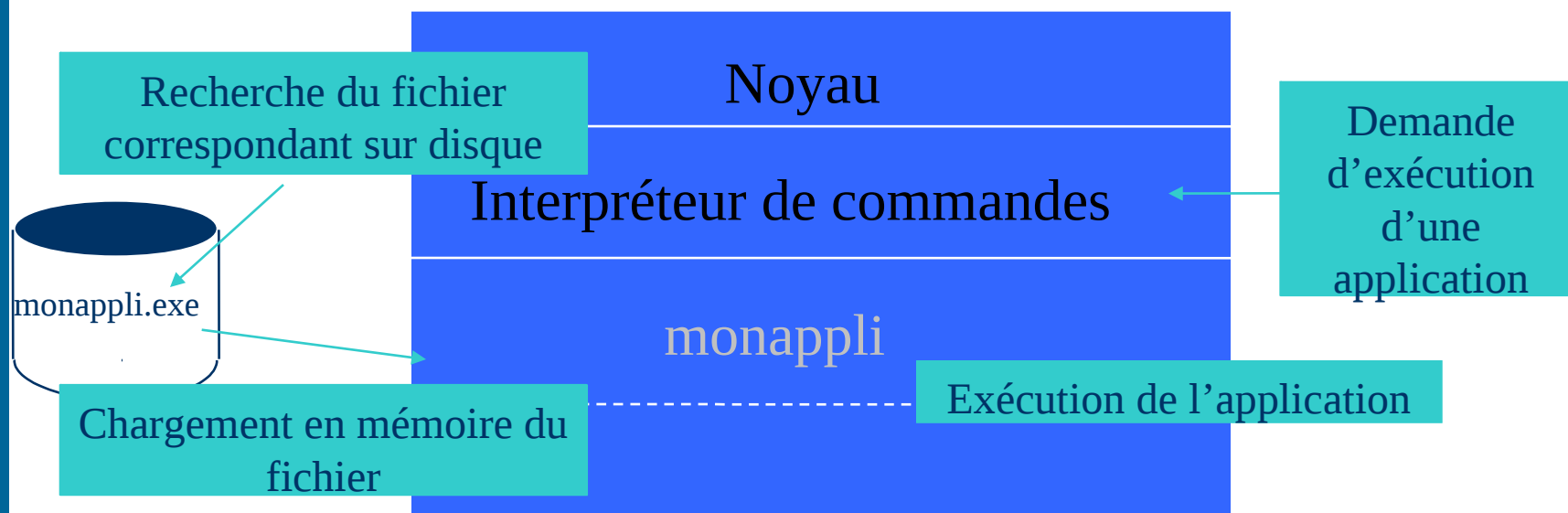
D'où vient le premier programme ?

- La mémoire "morte" (ROM) de l'UC contient un programme en langage machine : **l'amorce** (=bootstrap)
- Au démarrage de la machine cette amorce est automatiquement exécutée par le processeur
- Le rôle de l'amorce est de **charger** (= load) en mémoire les programmes du système d'exploitation situés sur le disque




Chargement des programmes

- Le noyau du système et au moins 1 interpréteur de commande restent **résidents** en mémoire centrale, les autres programmes sont chargés juste avant exécution



Systemes d'exploitation

- 
- Introduction : du matériel au logiciel
 - Quelques concepts « système » vus du côté de l'utilisateur
 - Rôles d'un système d'exploitation
 - Les interpréteurs de commandes
 - La gestion de fichiers
 - La protection des données



Concepts « système » vus côté utilisateur

● Instruction

- pour les langages machines, il s'agit d'un ordre de base compris par la machine (MOVE AX,2 OU ADD AX,BX)
- dans un langage évolué, une instruction correspond à un ensemble d'instructions du langage machine (print("coucou");)
- elles sont toujours composées d'un opérateur et des données permettant de réaliser l'opération

● Programme

- une suite d'instructions identifiée par un nom
- programme source / programme exécutable



Un processus

- = un programme en cours d'exécution

- Métaphore du cuisinier

- le cuisinier

- les ustensiles

- la recette de cuisine

- l'activité de préparation du plat

- le processeur

- les divers périphériques

- le programme

- le processus

- le système crée un processus qui exécute le programme demandé
- La plupart des systèmes permettent l'exécution de plusieurs processus simultanément



Les utilisateurs

- Les personnes qui travaillent sur l'ordinateur souvent identifiées par
 - un nom d'utilisateur (login)
 - un mot de passe
 - on parle d'accès à un **compte utilisateur**
- Une **session** de travail est la phase d'utilisation de la machine par un utilisateur
 - Dans un système mono-utilisateur :
session = démarrage/extinction de la machine
 - Dans un système multi-utilisateur :
session = connexion/déconnexion au système



Fichier

- = ensemble d'informations regroupées en vue de leur conservation et utilisation dans un système informatique : programmes, textes, images, son...
(= séquence de n octets)
- identifié par un nom (et souvent) une extension
- stocké en mémoire secondaire (disques durs, disquettes, bandes, cédérom...)
- les programmes sont stockés dans des fichiers
(les modules du système aussi)

Systemes d'exploitation

- Introduction : du matériel au logiciel
- Quelques concepts « système » vus du côté de l'utilisateur
- **Rôles d'un système d'exploitation (SE)**
- Les interpréteurs de commandes
- La gestion de fichiers
- La protection des données





Rôles d'un S.E.

Le SE peut être vu comme intermédiaire entre une couche application et les ressources physiques

- Pour la couche application, il rend un ensemble de services sur des concepts "logiques": fichiers, processus.
- Pour la couche physique:
 - il gère des ressources physiques
 - il contrôle et ordonne tous les dispositifs physiques de l'ordinateur (partage avec protection, performance)



Les services d'un S.E.

L'organisation en couches permet de

- masquer la complexité des opérations de pilotage du matériel (clavier, écran, souris, disques...)
- être indépendant de la technologie "courante"

Les services (**appels systèmes**) sont sous forme

- de fonctions appelables dans certains langages (C)
- de commandes soumises à un interpréteur de commandes

Pour un utilisateur courant, les services systèmes utiles sont ceux qui gèrent les fichiers.



Les principaux services systèmes

- Manipulation des fichiers
 - création, destruction, organisation des fichiers, lecture/écriture dans les fichiers, copie
- Contrôle des processus
 - chargement, exécution, interruption, terminaison
 - demandes d'allocation de ressources



Les applications pour programmeurs

- Interpréteurs de commandes
- Éditeurs de textes
- Compilateurs, éditeurs de liens, débogueurs
- Outils d'administration
 - ajout/suppression d'utilisateurs, surveillance, protection, quotas, comptabilité...
 - gestion des associations périphérique -pilote

Systemes d'exploitation

- Introduction : du matériel au logiciel
- Quelques concepts « système » vus du côté de l'utilisateur
- Rôles d'un système d'exploitation
- **Les interpréteurs de commandes**
- La gestion de fichiers
- La protection des données





Les interpréteurs de commandes (shell)

- Ce sont des programmes qui permettent à l'utilisateur d'interagir avec le système
 - exécution de programmes
 - gestion des fichiers
 - lecture des informations systèmes (version du système, date, personnes connectées, processus en cours...)
 - configuration de l'environnement de travail
- Ils peuvent être :
 - textuels : invite de commande (dos), bash, csh, tcsh (linux)...
 - graphiques : windows, finder du mac, kde (linux)..



Interpréteur textuel - langage de commandes

- En ouvrant une "application" terminal, on invoque un interpréteur (=shell) qui exécute, ligne à ligne, les commandes tapées, de syntaxe

nom_commande paramètres

Ex : `rm toto.txt` (détruit (remove) le fichier toto.txt)

- Les paramètres sont souvent des noms de fichiers
- Il y a plus de 100 commandes différentes



Interpréteur textuel - langage de commandes

- On peut écrire des programmes de commandes (macros) enregistrés dans un fichier qu'il suffira d'invoquer
- Des facilités pour éviter de taper du texte :
 - **copier-coller** pour modifier (=éditer) (voir plus loin)
 - **historique** (flèche haut) pour reprendre les commandes précédentes
 - **complétion automatique** des noms de fichiers (par tabulation) qui existent dans le chemin déjà spécifié



Interpréteur graphique - commandes

- Elles suivent une approche **objet** communes à toutes les applications :
 - on **désigne d'abord l'objet** sur lequel on va opérer (clic sur une icône de fichier)
 - on **applique ensuite une fonction** à cet objet (choix dans un menu, raccourci clavier)



Interpréteur graphique - commandes

- Édition (=modification) par (couper) copier coller

Ce concept utilise un tampon mémoire caché (buffer) : le presse-papiers

Couper

avant

après

Document
(répertoire)



Presse-papiers
(invisible)

???





Interpréteur graphique - commandes

- Édition (=modification) par (couper) copier coller

Ce concept utilise un tampon mémoire caché (buffer) : le presse-papiers

Copier

avant

après

Document



Presse-papiers

???





Interpréteur graphique - commandes

- Édition (=modification) par (couper) copier coller

Ce concept utilise un tampon mémoire caché (buffer) : le presse-papiers

Coller

avant

après

Document



Presse-papiers





Interpréteur graphique - commandes

- Copier/coller les fichiers ou répertoires
- Pour déplacer :
 - on coupe dans le répertoire initial
 - on colle dans le répertoire destinataire
- Pour détruire, on coupe
- Pour copier, on copie et on colle (plusieurs fois si nécessaire)



Interpréteur graphique - commandes

- Copier/coller marche aussi sur les textes

- Exemple : pour déplacer

j'étudierai , demain, sans faute, le cours.

sélectionner puis couper

→ j'étudierai le cours.

puis positionner le curseur

→ j'étudierai le cours. |

coller

→ j'étudierai le cours, demain, sans faute, .

corriger

→ j'étudierai le cours, demain, sans faute.



Interpréteur graphique - commandes

- Pour les déplacements, on a inventé le **glisser-déposer**
On **sélectionne** l'objet,
on le **déplace** en maintenant le bouton de souris enfoncé
jusqu'à **destination**.

Très utile pour déplacer des fichiers

- Le copier-coller ou glisser-déposer marchent pour un **groupe d'objets**



Interpréteur graphique - commandes

● Conclusion :

La plupart des opérations sur fichiers et répertoires
déplacer, copier, détruire
mais aussi

créer,renommer, changer les droits (voir plus loin)

se font facilement de manière graphique

Les **commandes textuelles** ne sont utiles que pour des
opérations **plus complexes.**



Le système d'aide

- Pour connaître la liste des paramètres et options d'une commande les interpréteurs disposent d'un système d'aide
 - commande **man** sous unix suivi du nom de la commande désirée (**help** en dos)
- Ces systèmes d'aide donnent plusieurs types d'informations
 - description succincte du rôle de la commande
 - syntaxe de la commande
 - fonctionnement détaillée de la commande



La variable \$PATH

- Les programmes exécutables sont invoqués par leur nom (ex : java ou gcc), mais normalement il faut leur nom complet, c.-à-d. le chemin depuis la racine (ex : /usr/bin/java)
- Le "shell" utilise une variable **\$PATH** qui regroupe tous les répertoires où les programmes généraux peuvent se trouver.
- C'est une macro exécutée à l'initialisation du shell qui définit \$PATH. C'est un administrateur qui vous l'installera. Si le \$PATH n'est pas bien défini, vous aurez le message : "java (ou gcc) command not found"

Systemes d'exploitation

- Introduction : du matériel au logiciel
- Quelques concepts « système » vus du côté de l'utilisateur
- Rôles d'un système d'exploitation
- Les interpréteurs de commandes
- **La gestion de fichiers**
- La protection des données





La gestion de fichiers

- Les fichiers sont l'entité de sauvegarde des informations d'un système informatique
- Il est donc important de pouvoir les retrouver facilement
 - Nécessité d'un système de classement des fichiers
 - Mécanisme de désignation basé sur ce classement
- ➔ Un système hiérarchique de boîtes imbriquées : les répertoires
- Il est également important de les protéger
 - Pour les préserver des suppressions intempestives
 - Pour préserver la confidentialité des informations qu'ils contiennent
- ➔ Un système de droits d'accès



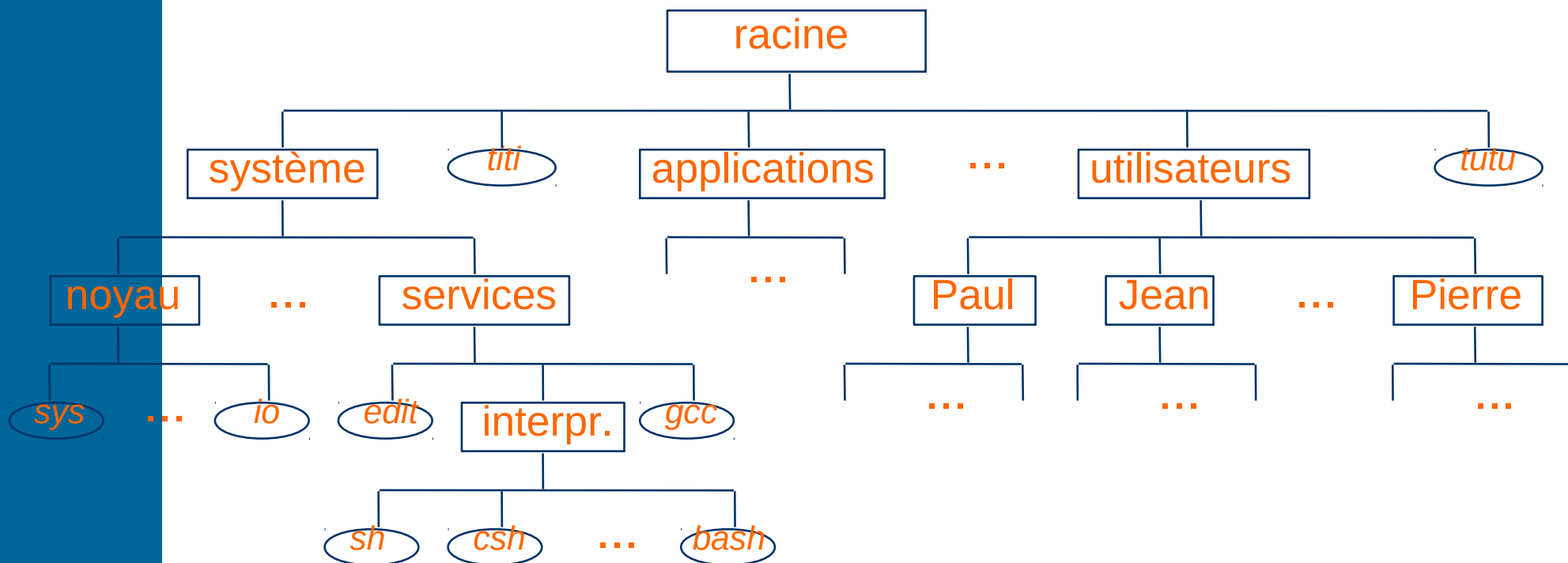
Organisation hiérarchique

- Les fichiers sont organisés dans une arborescence dont
 - La racine est le catalogue (ou répertoire principal)
 - Les nœuds internes sont des répertoires
 - Les feuilles sont des fichiers (ou des répertoires vides)
- Chaque répertoire contient des fichiers ou des sous-répertoires
 - On dit des répertoires qu'il contient que ce sont des **fils**
 - De même il est le **père** (ou répertoire parent) des répertoires qu'il contient



L'arborescence de fichiers

- Le répertoire racine n'a pas de père
- Tous les autres répertoires ont un père et un seul





Les commandes du SGF (Système de Gestion de Fichier)

- Création / destruction de répertoire
- Déplacement dans l'arborescence
 - le répertoire où l'on se situe est appelé **répertoire courant** et est désigné par **.**
 - on désigne par **..** le répertoire père
- Affichage du contenu d'un répertoire
- Création / destruction d'un fichier
- Affichage du contenu d'un fichier
- Copie d'un fichier ou d'une partie de l'arborescence
- Déplacement d'un fichier (ou d'une partie de l'arborescence)
- Renommage d'un fichier ou répertoire

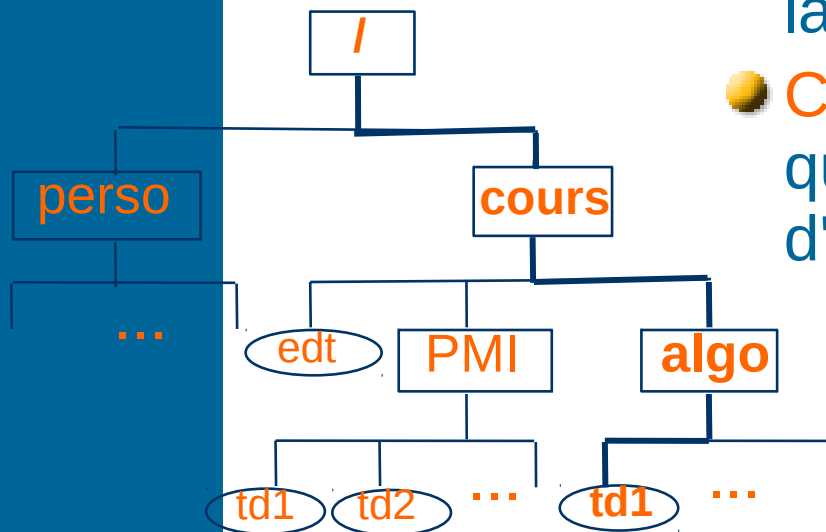


Désignation des fichiers et répertoires (1/2)

Deux modes de désignation sont possibles :

1. Par **référence absolue**

- Énumération des répertoires traversés de la racine jusqu'au fichier (ou répertoire)
 - Une telle référence démarre forcément par la racine / (\ chez microsoft)
 - **Cette désignation est unique** : il n'existe qu'un seul chemin de la racine à un nœud d'un arbre



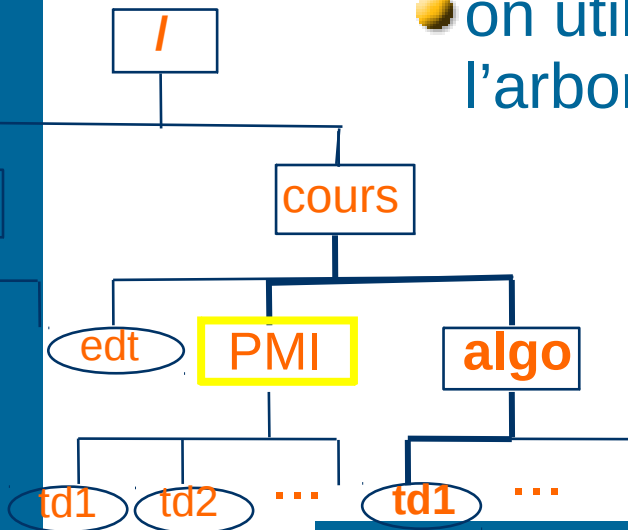
exemple : /cours/algo/td1



Désignation des fichiers et répertoires (2/2)

2. Par **référence relative**

- On utilise la notion de **répertoire courant** (ou de travail)
 - c'est le répertoire utilisé si aucun autre n'est spécifié (connu par **pwd** et modifié par **cd**)
- On énumère les déplacements à effectuer pour aller du répertoire courant jusqu'au fichier (ou répertoire) cible
 - on utilise le symbole **..** pour les remontées dans l'arborescence = le passage au répertoire père



exemple : ../algo/td1



Désignation de groupes de fichiers

- Mise en œuvre d'une technique de filtrage : on construit un filtre à l'aide de **caractères jokers**
 - * remplace n'importe quelle suite de caractères (même la suite vide) et ? remplace un caractère
 - Le filtre est appliqué à chacun des noms contenus dans le répertoire désigné par la commande
 - La commande est appliquée à tous les fichiers satisfaisant le filtre
- Exemple : **cp td* /cours/FLIN102** permet de copier tous les fichiers (et répertoires) du répertoire courant dont le nom commence par « td » dans le répertoire FLIN102



Fonctionnement du SGF

- Le SE fournit aux applications des **concepts** qui permettent :
 - d'abstraire les dispositifs physiques et donc
 - de dissimuler ses détails de fonctionnement interne
- Exemples :
 - un **fichier** pour abstraire la **mémoire secondaire**
 - un **processus** pour abstraire le **processeur**
 - une **mémoire virtuelle** pour abstraire la **mémoire physique**
 - un **flot de données** pour abstraire les **périphériques**



Fichiers vus de l'utilisateur

- Un fichier est un ensemble de données identifié par un nom et stocké en mémoire secondaire
- Pour désigner les fichiers (comme tout objet du système), l'utilisateur utilise des **identificateurs**
 - ➔ une chaîne de caractères dans un contexte d'utilisation
 - **nom** pour un fichier
 - **login** pour un utilisateur
 - ...



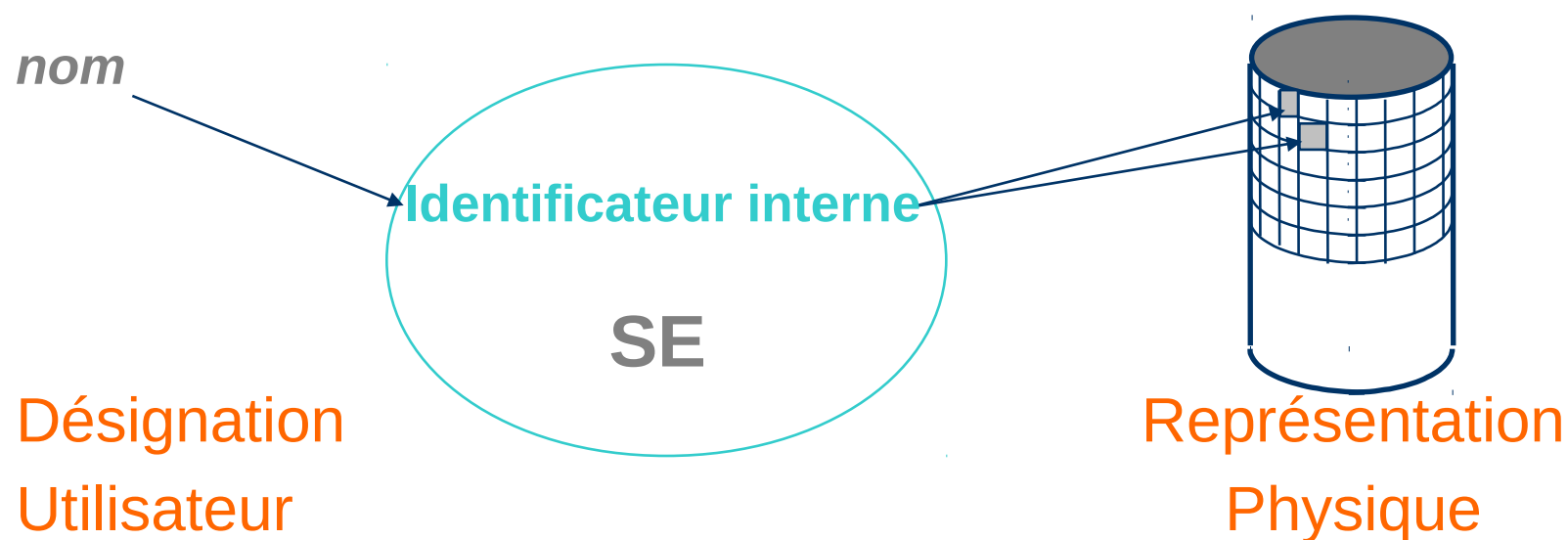
Fichiers vus du système

- Pour pouvoir manipuler un fichier, un SE doit connaître sa **représentation physique**
 - La ou les adresses des éléments physiques qui le supportent
 - La manière dont il est représenté sur le support
 - La ou les actions physiques permettant de modifier cet objet
- La représentation dépend du support physique (disque dur, CD-ROM, bandes magnétiques...)



Liaison nom<->représentation

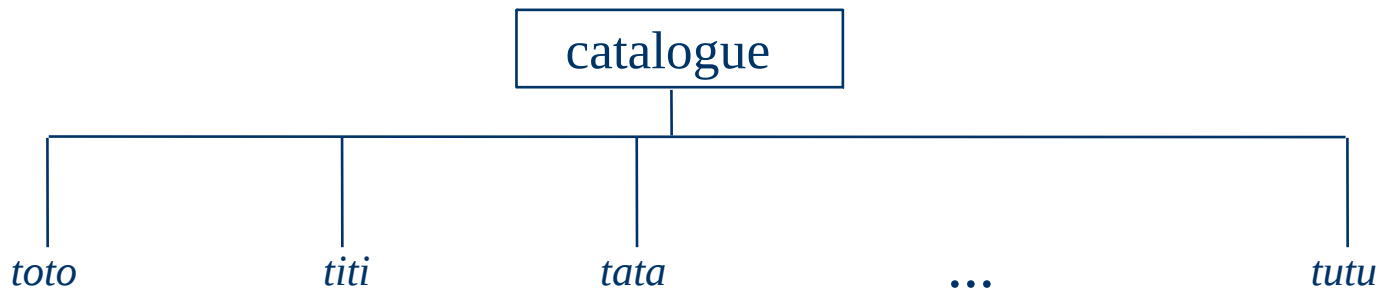
- C'est à la charge du SE d'établir la **liaison** entre le nom utilisé par l'utilisateur et la représentation physique du fichier





Organisation du catalogue

- Avec un catalogue à un seul niveau

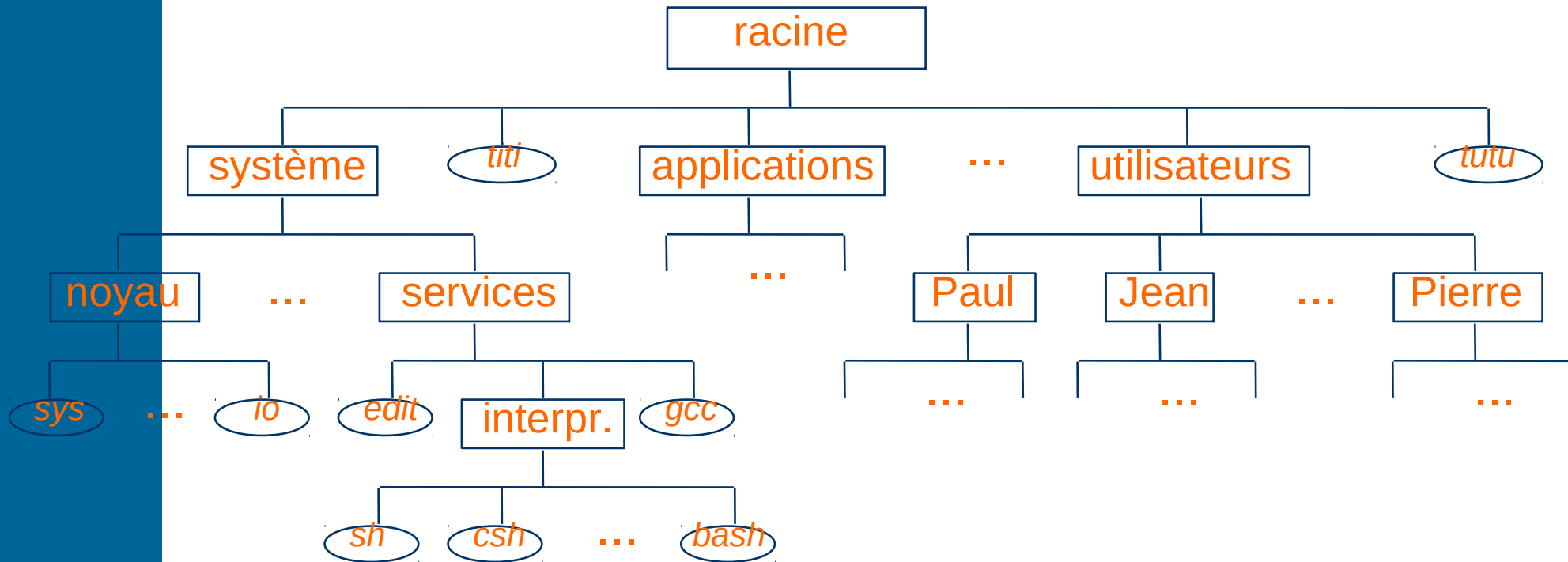


- les fichiers doivent tous avoir des noms différents
- pas de classement par rubrique possible
- les fichiers des divers utilisateurs sont mélangés
- la recherche d'un fichier impose de parcourir l'ensemble des fichiers du catalogue
- D'où la nécessité d'une **organisation arborescente en répertoires** du catalogue



L'arborescence de fichiers

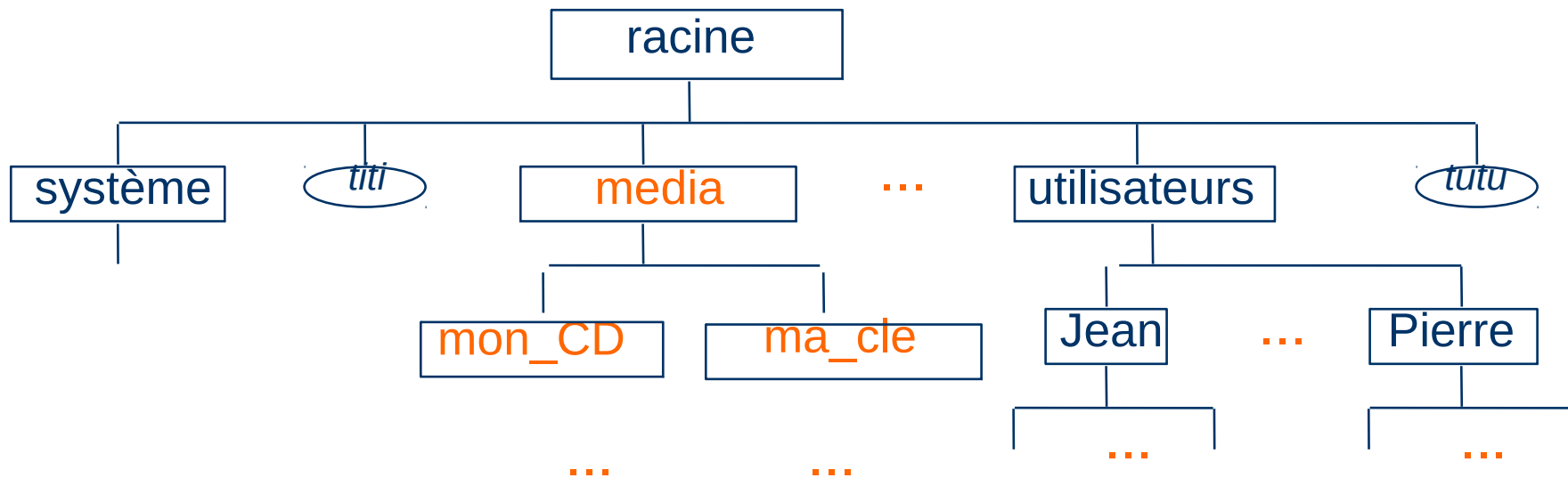
- On aboutit ainsi à une arborescence





Les supports amovibles et le SGF

- Un support amovible (zip, cd,dvd,clé USB) ou "réseau" contient aussi une arborescence de fichiers qui est greffée (au montage) sur l'arborescence générale



Linux : dans le répertoire "media"

Windows : dans le répertoire "poste de travail"

Systemes d'exploitation

- Introduction : du matériel au logiciel
- Quelques concepts « système » vus du côté de l'utilisateur
- Rôles d'un système d'exploitation
- Les interpréteurs de commandes
- La gestion de fichiers
- La protection des données





La protection

- Le problème : permettre un partage de fichiers entre utilisateurs mais aussi une protection
- L'ensemble des opérations autorisées sur un objet constitue ses **droits d'accès** (donnés par le propriétaire)
- Les droits peuvent évoluer dans le temps (ex : un corrigé lisible par les étudiants)



Les droits d'accès Linux

- Trois types d'accès sont possibles (**rw**x)
 - l'accès en lecture (**r**ead)
 - permet de lire le contenu d'un fichier
 - permet de lister le contenu d'un répertoire
 - l'accès en écriture (**w**rite)
 - permet de modifier un fichier
 - permet d'ajouter ou de supprimer des fichiers et sous-répertoires dans un répertoire
 - l'accès en exécution (**e**xecute)
 - si le fichier est un programme exécutable : permet de l'exécuter
 - si répertoire : permet de l'ouvrir



Lecture des droits **ls -l** (1/2)

La commande **ls-l** associe à chaque fichier 10 caractères:

- le 1^{er} = nature du fichier (**d** pour répertoire et **tiret (-)** pour fichier simple)
- les 9 suivants = droits attribués à ce fichier (3 pour le propriétaire, 3 pour le groupe, 3 pour les autres)
 - Chaque groupe de 3 : 1^{er} caractère = droit en lecture
2^e caractère = droit en écriture
3^e caractère = droit en exécution
- Pour chaque droit, si sa lettre apparait, le droit est donné, si il y a un tiret (-), le droit est refusé.



Lecture des droits **ls -l** (2/2)

● Exemple:

Type droits		prop	groupe	taille	date_modification	nom
-rw-rw-r--	1	prof1	prof	502	sept 2 9:40	fic1.txt
-rwx--x--x	1	prof1	prof	2055	aout 3 8:03	monprog
drwxr-x---	2	prof1	prof	4096	sept 1 7:35	FLIN102

- fic1.txt est accessible en lecture et écriture par prof1 et les membres du groupe prof et en lecture pour tous les autres
- monprog est accessible en lecture, écriture et exécution par prof1 et en exécution seulement pour tous les autres
- FLIN102 est un répertoire qui est listable (droit r) et traversable (droit x) par les membres du groupe prof ; prof1 peut en plus le modifier (droit w) ; les autres n'ont aucun droit