POLYTECH DIJON

SHELL

SYSTÈME D'EXPLOITATION

charles.meunier@u-bourgogne.fr

OBJECTIFS

Compétences attendues à la fin de ce module :

DÉFINIR ce qu'est un **SYSTÈME D'EXPLOITATION**

COMPRENDRE le fonctionnement d'un **SYSTÈME DE FICHIERS**

RÉALISER l'**ADMINISTRATION DE BASE** d'un système Linux

ÉCRIRE ET **EXÉCUTER** DES SCRIPTS SHELL

charles.meunier@u-bourgogne.fr

Présentation du module

ORGANISATION DES ENSEIGNEMENTS

15 **HEURES**	3 **CM**	2 **TD**	2 **TP**
	2 **ÉPREUVES**	Théorique : **24 octobre 2025** Pratique : **dernière séance de TP**	

charles.meunier@u-bourgogne.fr

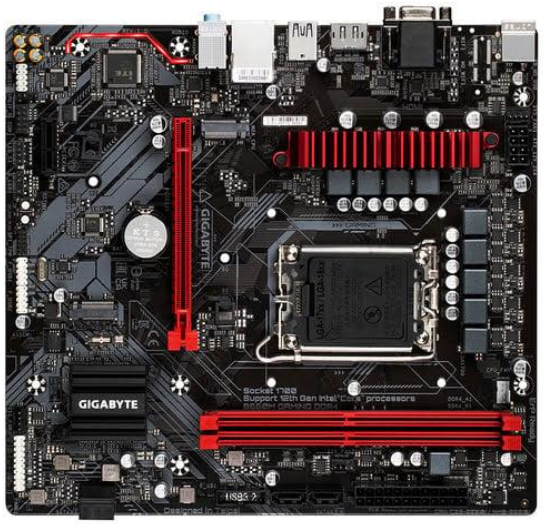
POLYTECH DIJON

SYSTÈME D'EXPLOITATION

charles.meunier@u-bourgogne.fr

LES COMPOSANTS D'UN ORDINATEUR

CARTE GRAPHIQUE

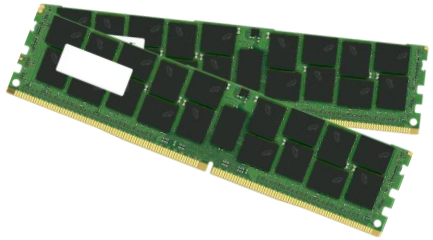


CARTE MÈRE

MICROPROCESSEUR

DISQUE DUR

ALIMENTATION

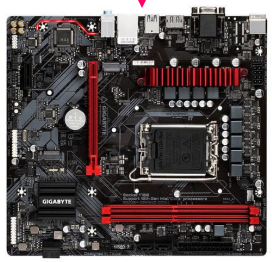


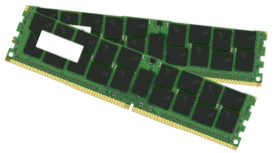
MÉMOIRE VIVE (RAM)

charles.meunier@u-bourgogne.fr

UNE INTERFACE AVEC LE MATÉRIEL

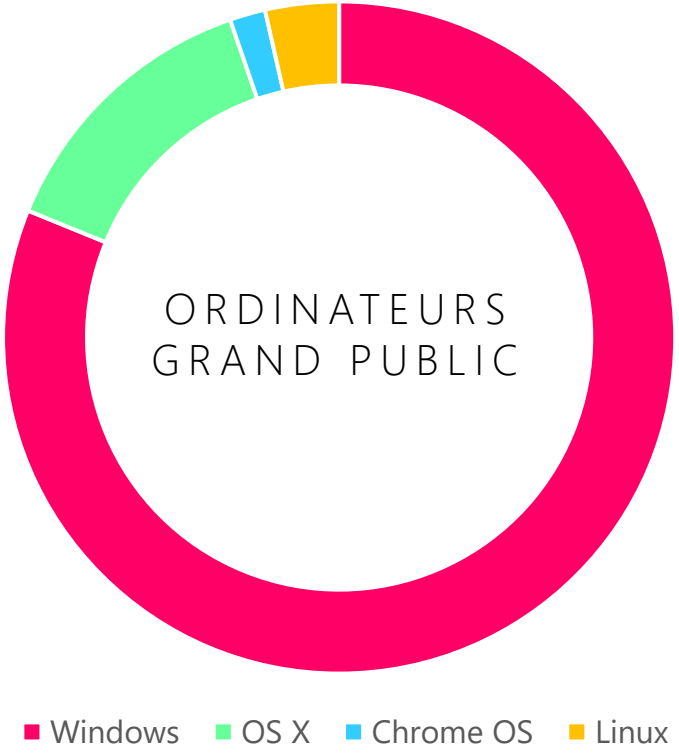
SYSTÈME D'EXPLOITATION

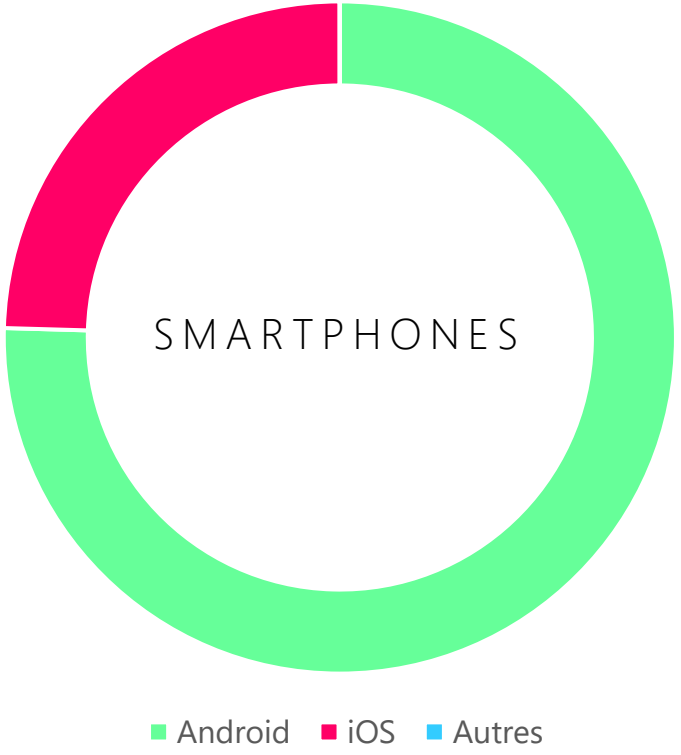




charles.meunier@u-bourgogne.fr

LES PLUS CONNUS





source : https://gs.statcounter.com/os-market-share

LARGE ÉCOSYSTÈME

POLYTECH DIJON

PROPEROS OS
PlayStation 5

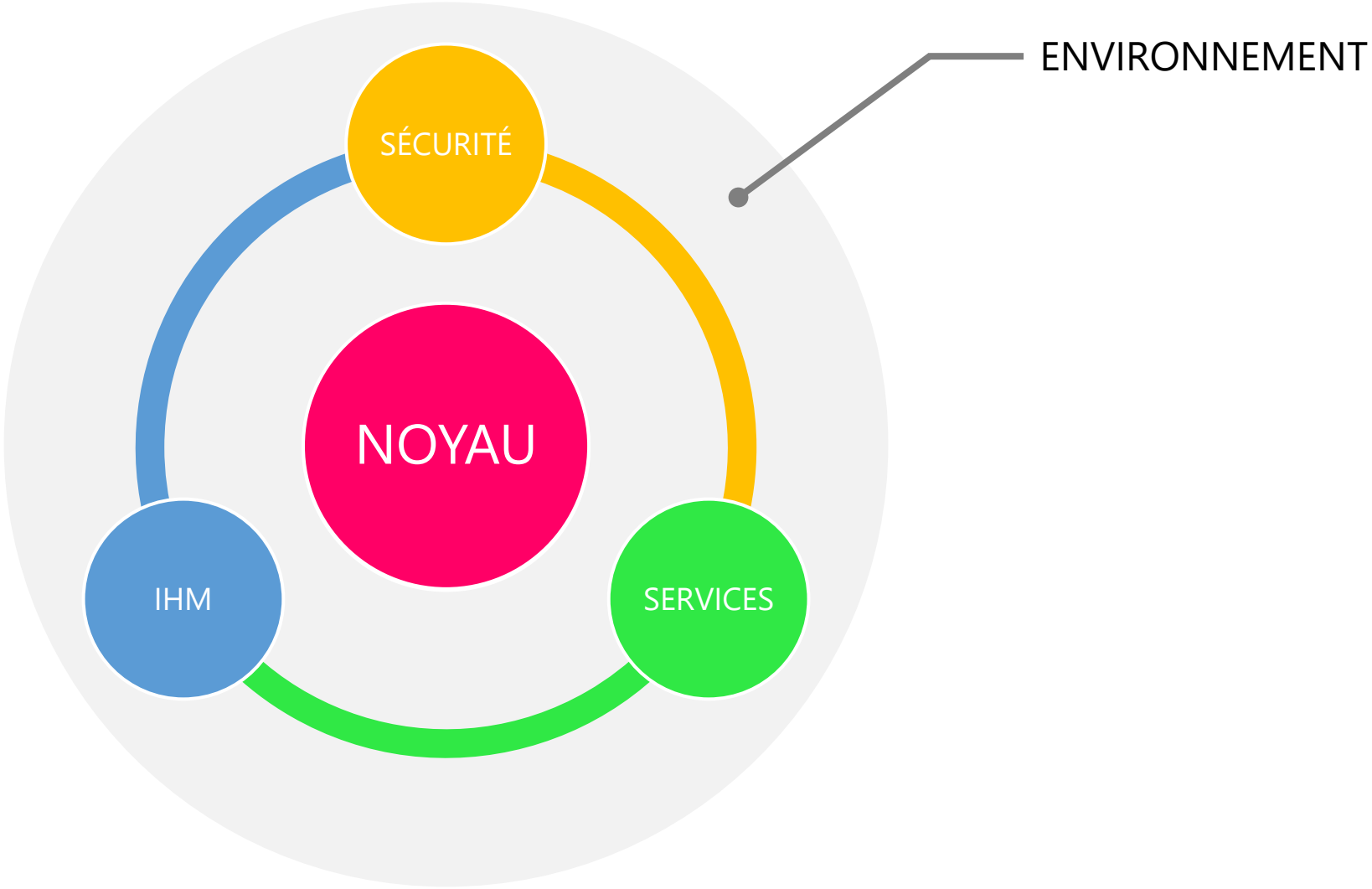
IBM z/OS
Mainframe IBM

ESXi
Hyperviseur

...ET BIEN D'AUTRES...

charles.meunier@u-bourgogne.fr

NOYAU ET ENVIRONNEMENT



charles.meunier@u-bourgogne.fr

NOYAU

Le noyau est **LE COMPOSANT CENTRAL** du système d'exploitation

Ses missions :

- Interface avec le matériel (processeur, mémoire, périphériques)
- Gestion de la mémoire vive
- Gestion de l'ordonnanceur
- Gestion des périphériques
- Gestion du système de fichiers

charles.meunier@u-bourgogne.fr

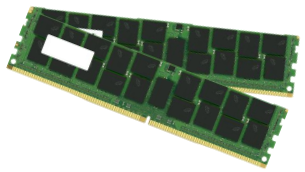
POLYTECH DIJON

MÉMOIRE

charles.meunier@u-bourgogne.fr

QUE REPRÉSENTE LA MÉMOIRE ?

MÉMOIRE DE MASSE
volumineuse et lente



MÉMOIRE PRINCIPALE
RAM

MÉMOIRE CACHE
rapide mais restreinte

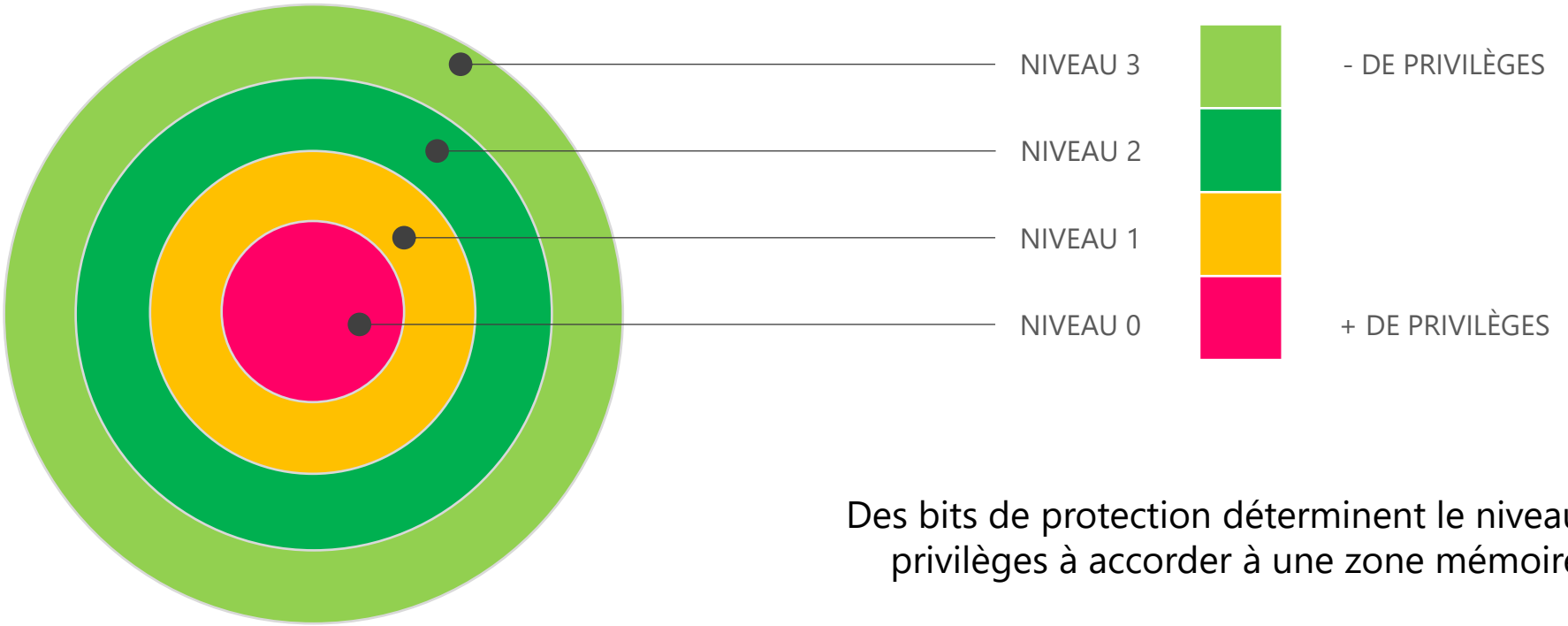
ADRESSES	VALEURS
0x00000001	'H'
0x00000002	'i'
0x00000003	'!'
...	...
0xFFFFFFFE	42
0xFFFFFFFF	1337

Vision simplifiée de la mémoire

charles.meunier@u-bourgogne.fr

SÉCURISATION DE LA MÉMOIRE PRINCIPALE

La mémoire principale est "cloisonnée" en espaces hiérarchisés en termes de sécurité : les **ANNEAUX DE PROTECTION**

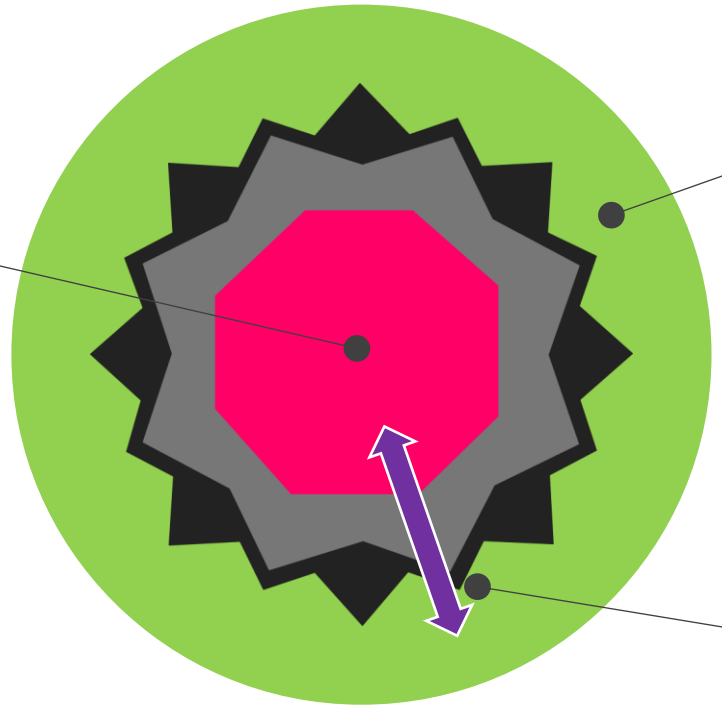


Des bits de protection déterminent le niveau de privilèges à accorder à une zone mémoire.

charles.meunier@u-bourgogne.fr

NOYAU

Aujourd'hui, la plupart des systèmes d'exploitation utilisent deux niveaux de privilèges :



MÉMOIRE NOYAU

Seuls les processus noyau peuvent y être chargés et exécutés

MÉMOIRE UTILISATEUR

Espace mémoire pour tous les autres processus

APPELS SYSTÈMES

Les processus en mémoire utilisateur ne peuvent pas accéder directement à la mémoire noyau.

Ils doivent passer par des appels systèmes

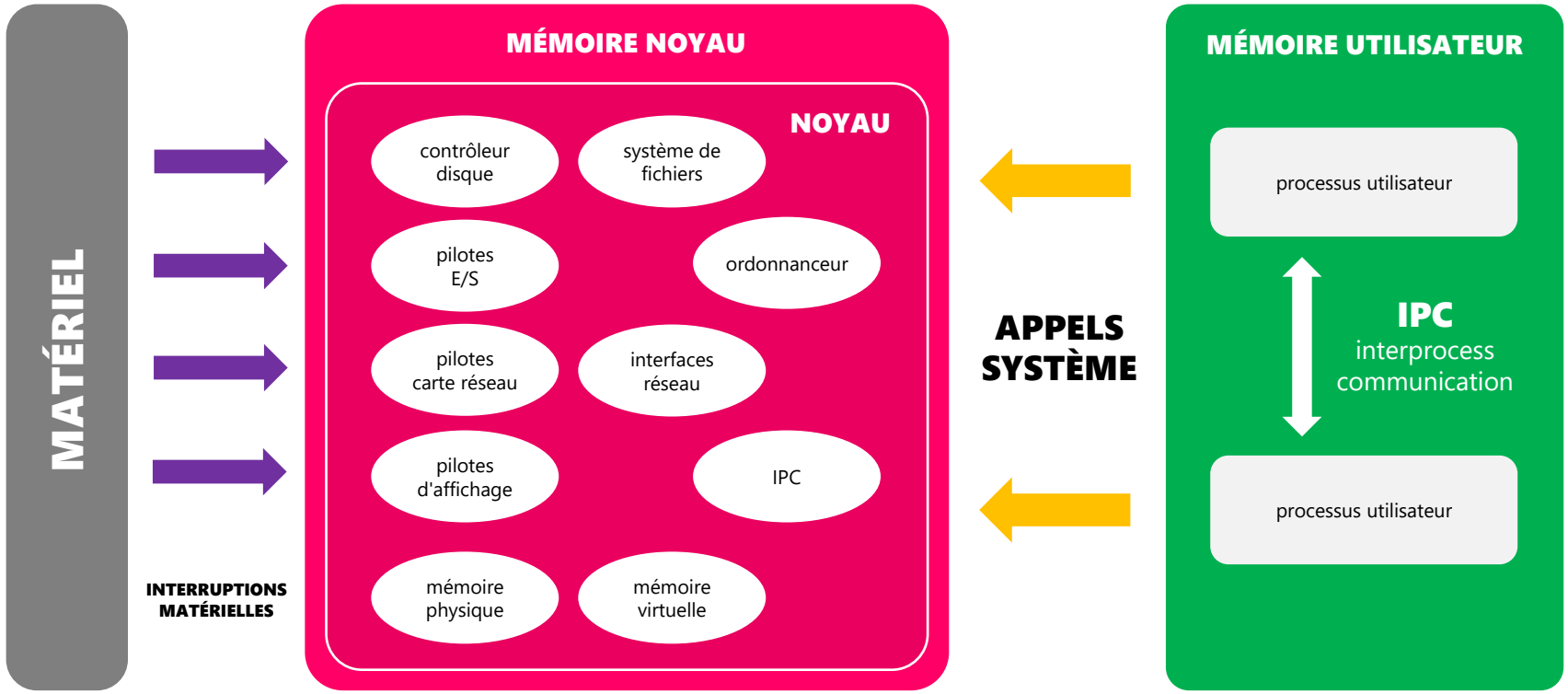
POLYTECH DIJON

NOYAU

charles.meunier@u-bourgogne.fr

MONOLITHIQUE

Toutes les tâches du noyau sont regroupées au sein d'**UN SEUL PROCESSUS** situé dans la mémoire noyau.

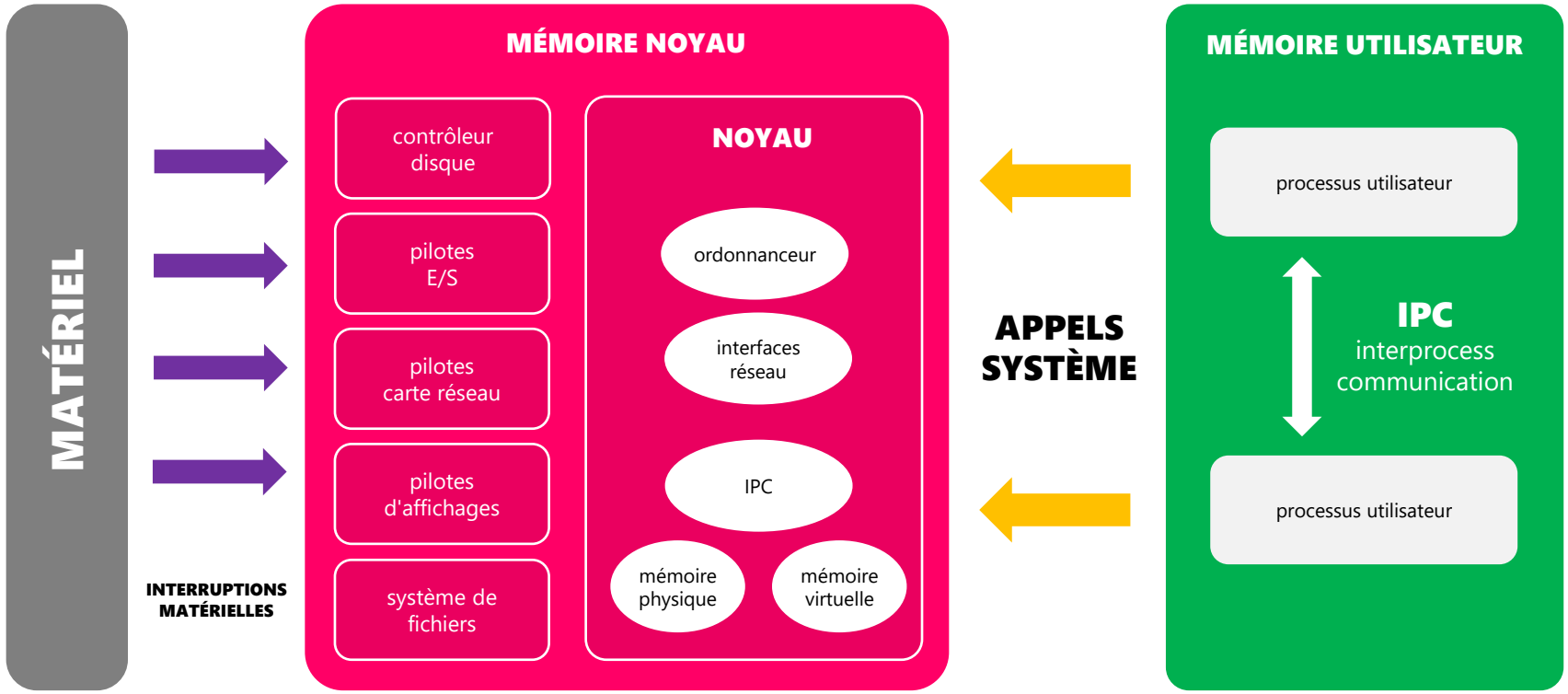


AVANTAGES : Très performant

INCONVENIENTS : Complexe à maintenir + Risque d'instabilité

MONOLITHIQUE MODULAIRE

Le code des modules est indépendant du code noyau. Les modules sont chargés dans le noyau en fonction des besoins.

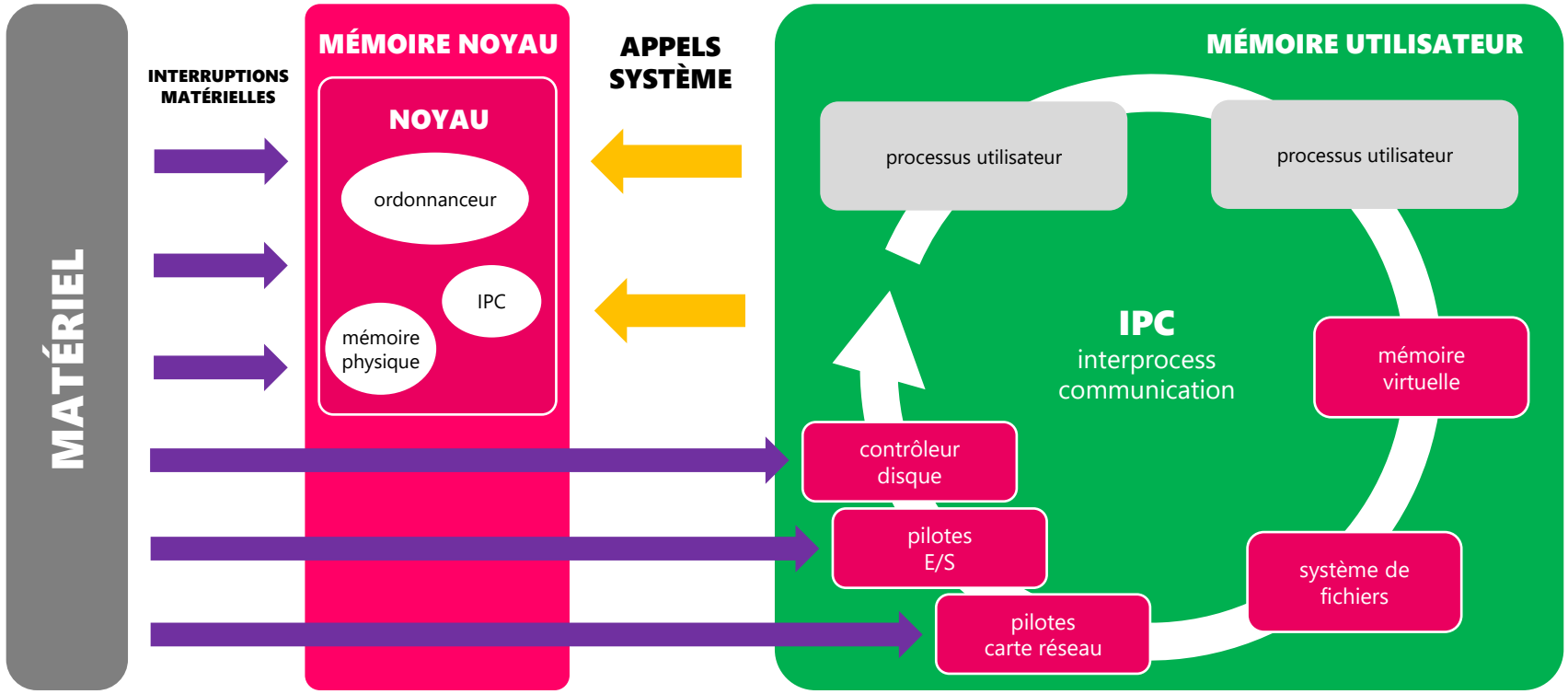


AVANTAGES : Très performant + Maintenance facilité par l'architecture modulaire

INCONVENIENTS : Toujours des risques au niveau de la stabilité

MICRO-NOYAU

Seules les fonctions critiques restent dans le noyau. Les autres fonctions sont exportées comme services chargés dans la mémoire utilisateur.

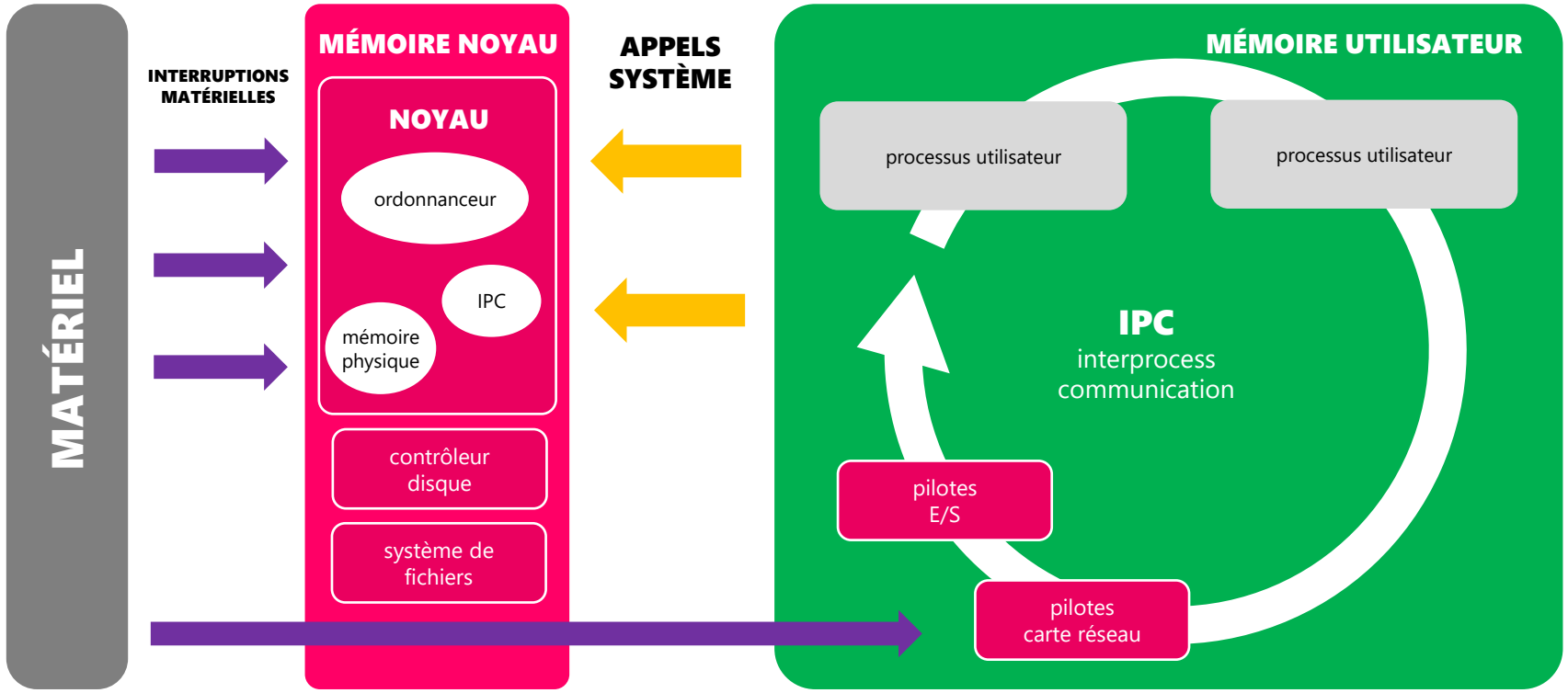


AVANTAGES : La stabilité du système (le crash d'un service en mémoire utilisateur n'a pas d'impact sur le noyau).

INCONVENIENTS : Baisse des performances due à la multiplication des IPC.

HYBRIDE

Même principe que le micro-noyau mais les fonctions qui génèrent beaucoup d'appels systèmes sont réintégrées au noyau.



AVANTAGES : Améliore grandement les performances par rapport au micro-noyau.

INCONVENIENTS : Reste moins performant qu'une architecture monolithique.

COMPARATIF

A VOUS DE JOUER !

NOYAU	AVANTAGES	INCONVÉNIENTS	EXEMPLES
Monolithique	Très performant	Difficile à maintenir Risque d'instabilité	Premières versions de Linux, z/OS
Modulaire	Très performant	Risque d'instabilité	Linux modernes
Micro-noyau	Stable	A priori moins performant	QNX
Hybride	Stable	Moyennement performant	Windows 11 (NT), Mac OS

POLYTECH DIJON

PROCESSUS

charles.meunier@u-bourgogne.fr

PROGRAMME VS EXÉCUTABLE

programme.cpp

```
#include <iostream>

int main()
{
    int a = 5;
    int b = 8;

    std::cout << a + b << std::endl;
}
```

PROGRAMME

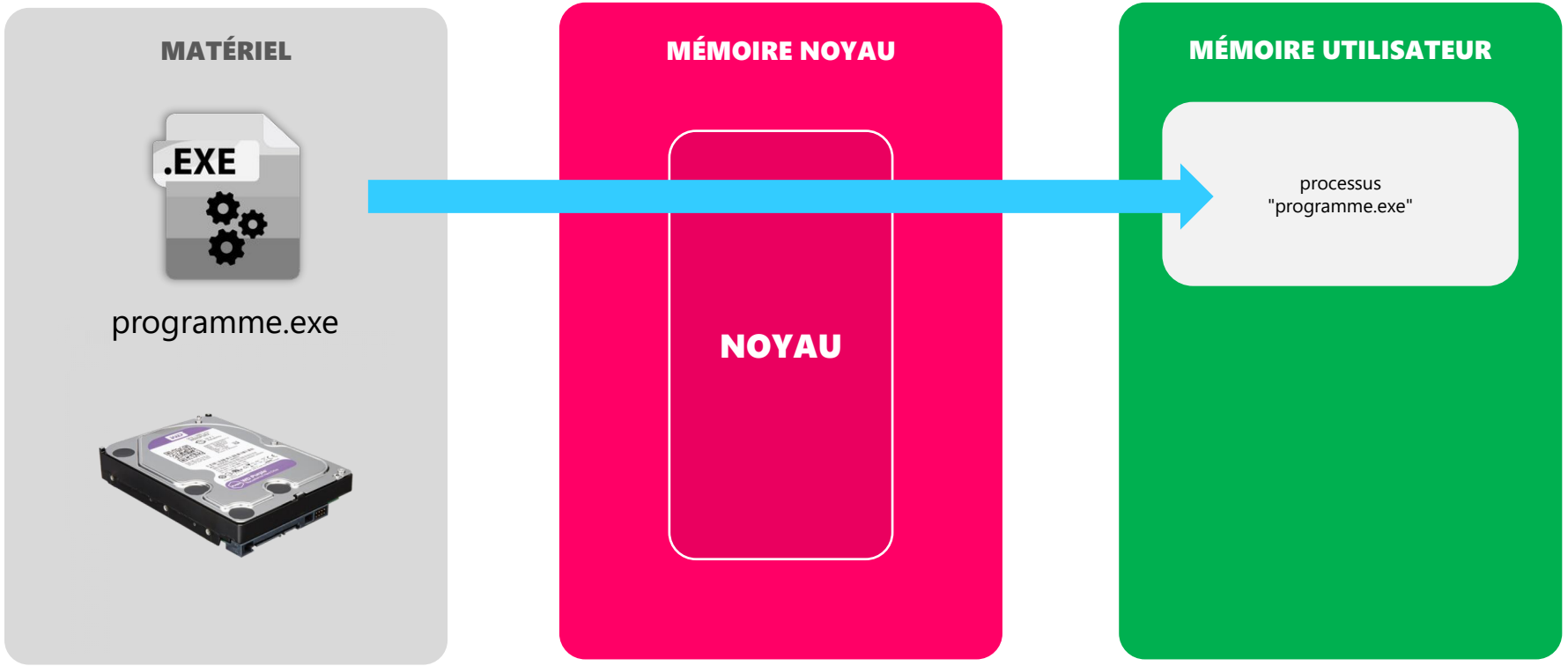
programme.exe

```
pushq   %rbp
movq    %rsp, %rbp
subq    $16, %rsp
movl    $5, -4(%rbp)
movl    $8, -8(%rbp)
movl    -4(%rbp), %edx
movl    -8(%rbp), %eax
addl    %edx, %eax
movl    %eax, %esi
movl    $_ZSt4cout, %edi
call    _ZNSolsEi
movl    $_ZSt4endlIcSt11char_traitsIcEERSt13basic_ostreamIT_T0_ES6_,
%esi
movq    %rax, %rdi
call    _ZNSolsEPFRSoS_E
movl    $0, %eax
leave
ret
```

EXÉCUTABLE

charles.meunier@u-bourgogne.fr

EXÉCUTABLE VS PROCESSUS

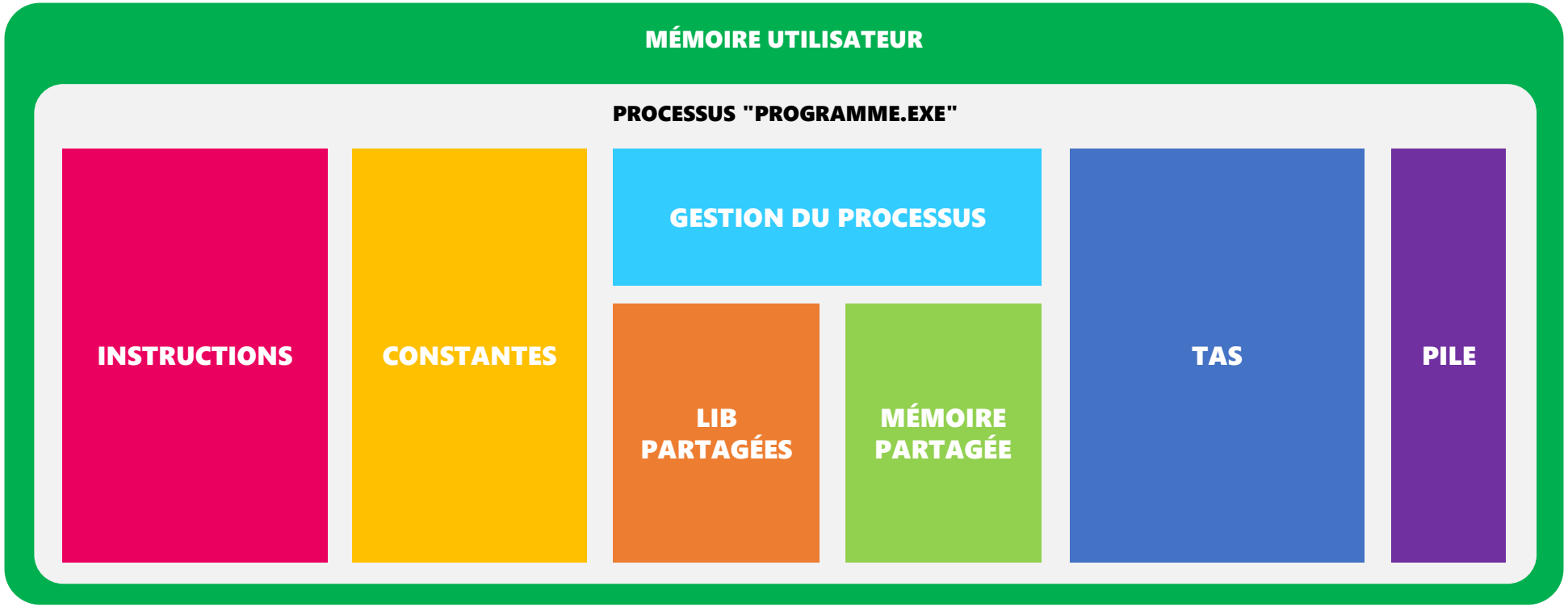


Le système d'exploitation crée un **PROCESSUS EN MÉMOIRE PRINCIPALE**

à partir de l'**EXÉCUTABLE CHARGÉ DEPUIS LA MÉMOIRE DE MASSE**

ESPACE MÉMOIRE

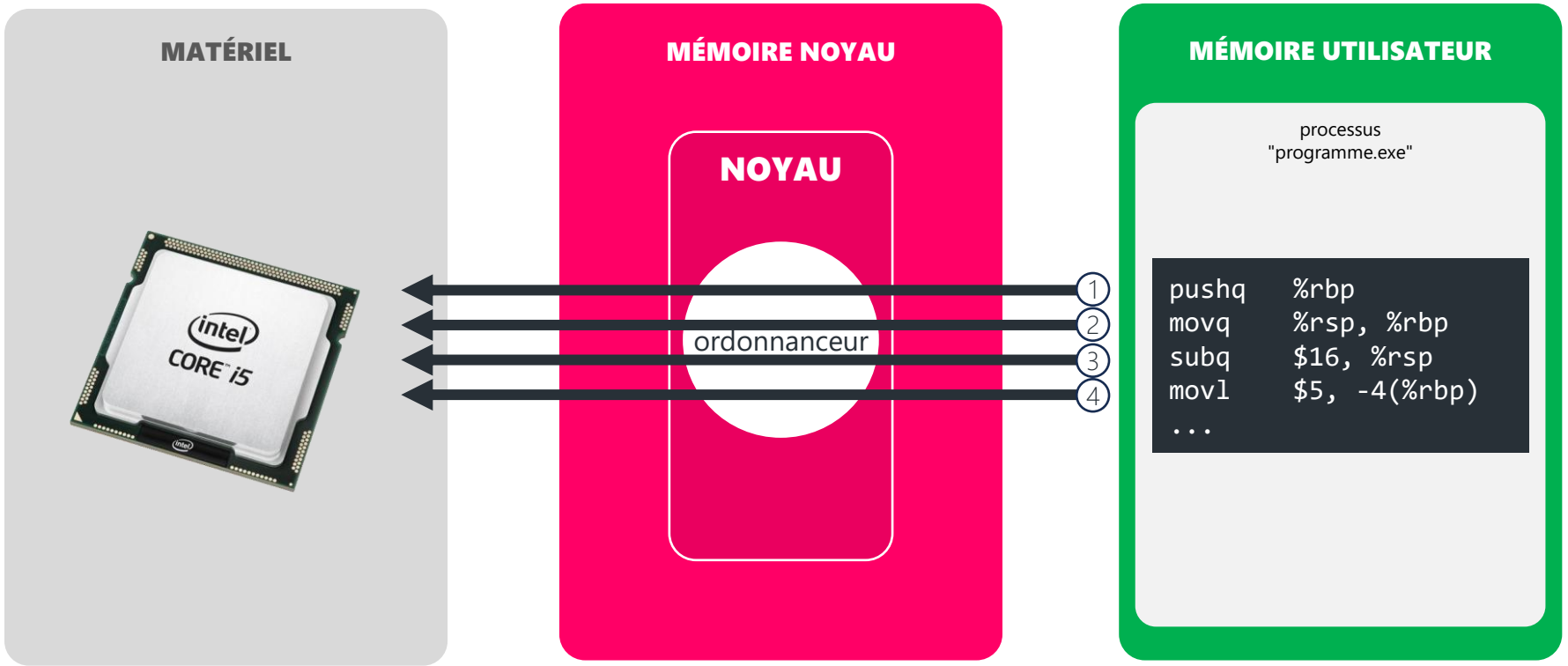
Chaque processus dispose de son propre espace mémoire : l'**ESPACE D'ADRESSAGE**.



Un processus **NE PEUT PAS** accéder à l'espace d'adressage d'un autre processus en dehors de la mémoire partagée.

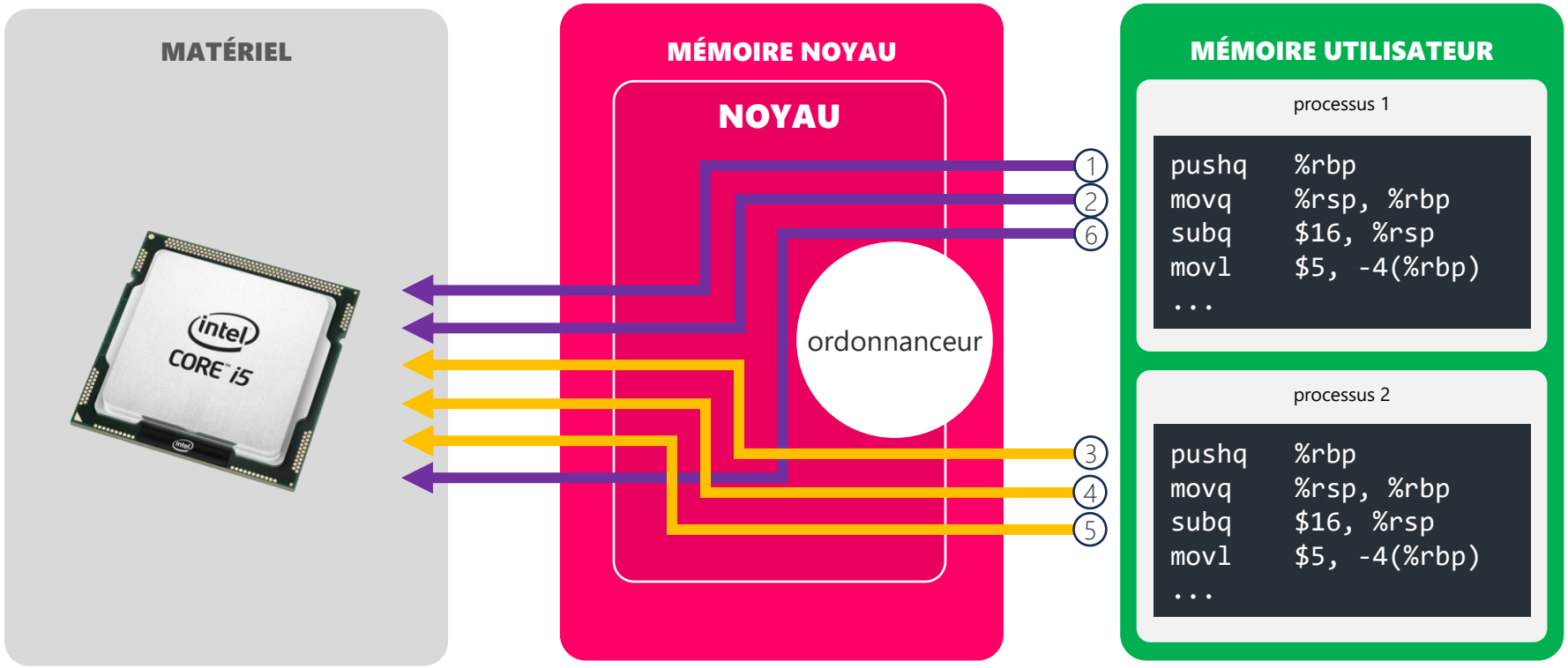
ORDONNANCEUR

Une fois le processus chargé en mémoire, l'ordonnanceur gère l'exécution de ses instructions



MULTITÂCHES

L'ordonnanceur est en charge du multitâche, c'est à dire partager le temps processeur entre les différents processus pour donner à l'utilisateur l'illusion d'un traitement simultané.



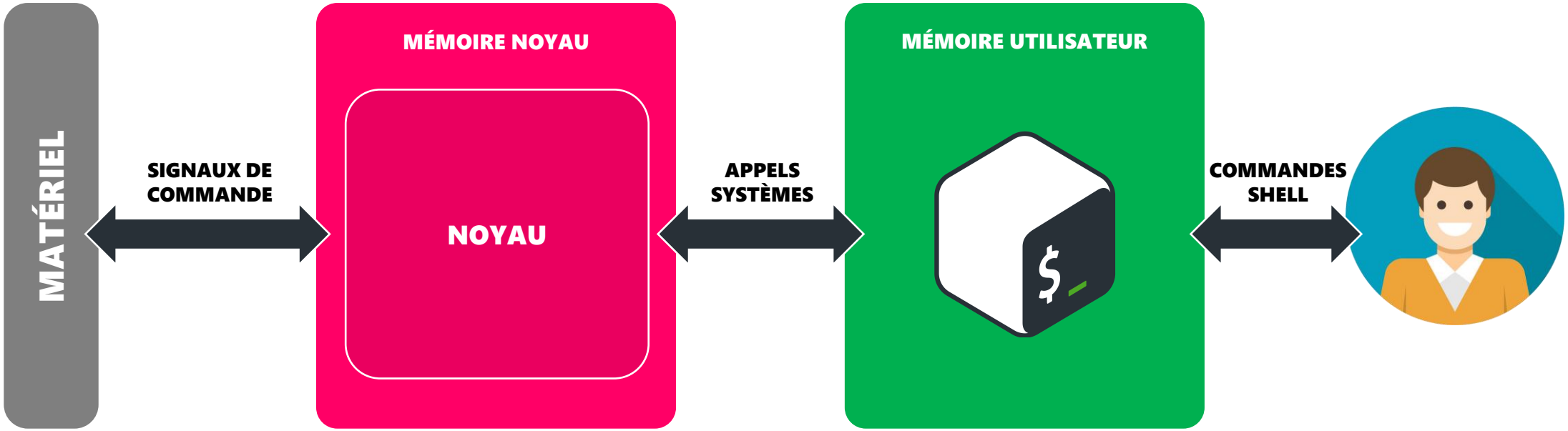
POLYTECH DIJON

SHELL

charles.meunier@u-bourgogne.fr

DE QUOI S'AGIT IL ?

Le terme SHELL désigne une interface en ligne de commande d'un système d'exploitation.
Il permet à un utilisateur d'interagir avec le système d'exploitation, et donc, indirectement, avec l'ordinateur.



Le SHELL interprète les commandes saisies par l'utilisateur et les traduit en appels système.

COMMENT Y ACCÉDER

LINUX

OS X

TERMINAL

WINDOWS

INVITE DE COMMANDE
(CMD)

POWERSHELL

Windows 11 intègre un nouveau **TERMINAL** permettant d'accéder à tous les outils en lignes de commande

charles.meunier@u-bourgogne.fr

PLEINS DE SHELLS

Pour chaque système d'exploitation, il existe plusieurs shells, chacun apportant sa syntaxe et ses fonctionnalités.

WINDOWS

CMD

Shell issu de MS DOS aux fonctionnalités.

PowerShell

Shell basé sur la programmation orientée objets et sur le Framework ,NET de Microsoft. Disponible également sous OS X et Linux.

Linux

BASH (Bourne Again SHell)

Shell par défaut sur la plupart des distributions.

ZSH

Shell personnalisable, permettant de manipuler des nombres décimaux et offrant une complétion automatique avancée.

FISH (Friendly Intercative SHell)

Shell conçu pour les débutants. Syntaxe intuitive mais très différente de BASH.

DASH

Shell minimaliste disponible sur Debian

charles.meunier@u-bourgogne.fr

PROMPT

POLYTECH DIJON

Chaque shell est identifiable par son prompt qui fournit plus ou moins d'informations

BASH

```
cmeunier@LAPTOP-B3V8QTNO:~$
```

FISH

```
cmeunier@LAPTOP-B3V8QTNO ~>
```

DASH

```
$
```

ZSH

```
LAPTOP-B3V8QTNO%
```

POWERSHELL

```
PS C:\Users\cmeunier>
```

CMD

```
C:\Users\cmeunier>
```

SYNTAXE D'UNE COMMANDE

POLYTECH DIJON

Une commande est une chaine de caractères interprétée par le shell pour réaliser une action auprès du système d'exploitation.

Nom de la commande : la syntaxe commence par le nom de la commande à invoquer

```
ls
```

Options : puis on retrouve généralement les options dont le nom est précédé d'un ou deux tirets.

```
ls -a -l
```

```
ls -al
```

```
ls --all -l
```

Certaines options nécessitent une valeur :

```
find -name "*.cpp"
```

Paramètres : enfin, certaines commandes prennent des paramètres en entrée

```
mkdir Documents
```

charles.meunier@u-bourgogne.fr

DOCUMENTATION

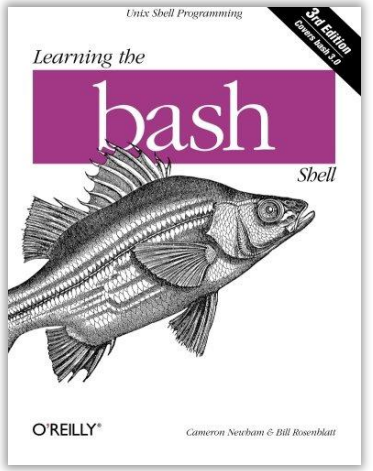
Pour savoir comment utiliser une commande :

LE MANUEL

```
man find
```

L'AIDE FOURNIE PAR LA COMMANDE ELLE-MÊME

```
find --help
```



COMMANDES DE BASE

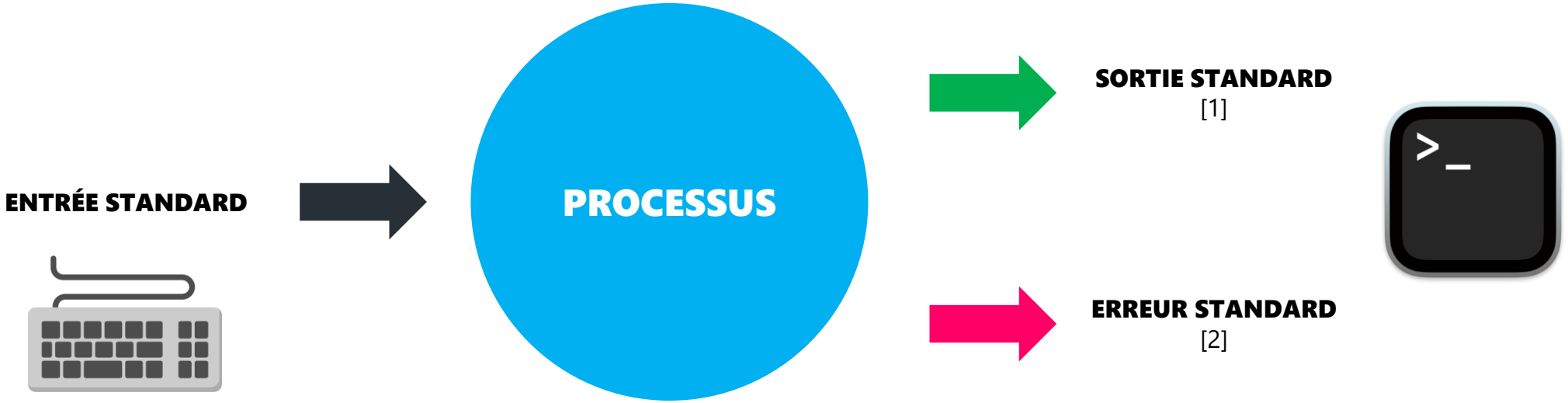
POLYTECH DIJON

REVOIR LES TPs DE SOUTIEN LINUX

https://lamarmotte.info

charles.meunier@u-bourgogne.fr

ENTRÉES/SORTIES STANDARDS



charles.meunier@u-bourgogne.fr

REDIRECTION DES SORTIES

Par défaut, une commande affichera son résultat dans la console.

Il est possible de rediriger le résultat d'une commande **dans un nouveau fichier**

```
ls -al > monFichier
```

ou

```
ls -al 1> monFichier
```

Attention : si monFichier existe, il sera écrasé

Il est possible de rediriger le résultat d'une commande **à la fin d'un fichier existant**

```
ls -al >> monFichier
```

ou

```
ls -al 1>> monFichier
```

Si monFichier n'existe pas, il sera créé

Il est possible de rediriger le résultat d'une commande **vers une autre commande**

```
ls -al | grep -E "^-.{0,8}x" | awk '{print $9}'
```

Le résultat de la commande **`ls`** est traité par la commande **`grep`** dont le résultat est à son tour traité par la commande **`awk`**

REDIRECTION DES ERREURS

Par défaut, lorsqu'une erreur se produit lors de l'exécution d'une commande, le message sera affiché dans la console.

Il est possible de rediriger le message d'erreur d'une commande **dans un nouveau fichier**

```
ls -al dosserInconnu 2> logError
```

Attention : si logErreur existe, il sera écrasé

Il est possible de rediriger le message d'erreur d'une commande **à la fin d'un fichier existant**

```
ls -al dossierInconnu 2>> logError
```

Si monFichier n'existe pas, il sera créé

Il est possible de rediriger le message d'erreur d'une commande **vers la même sortie que la sortie standard**

```
ls -al dossierInconnu 1>> log 2>&1
```

Si la commande se déroule correctement, le résultat est envoyé dans le fichier log.

Si une erreur survient, le message d'erreur est aussi envoyé dans le fichier log.

REDIRECTION DES ENTRÉES

Le clavier est l'entrée sélectionnée par défaut lorsqu'une commande à besoin d'information au cours de son exécution.

```
sudo passwd username
New password:
```

Ici, la commande attend la saisie du nouveau mot de passe par l'utilisateur

Il est possible de faire d'un fichier l'entrée standard d'un processus :

```
sudo passwd username < defaultPassword.txt
New password: Retype new password: passwd: password updated successfully
```

Ici, la commande a pioché les informations dont elle avait besoin dans le fichier defaultPassword.txt

Il est possible de faire d'un "HERE DOCUMENT" l'entrée standard d'un processus :

```
sudo passwd username << END
toto21
toto21
END
New password: Retype new password: passwd: password updated successfully
```

Ici, la commande a pioché les informations dans le texte délimité par "END"

POLYTECH DIJON

LA PROCHAINE FOIS

Systèmes de fichiers et premiers scripts

charles.meunier@u-bourgogne.fr