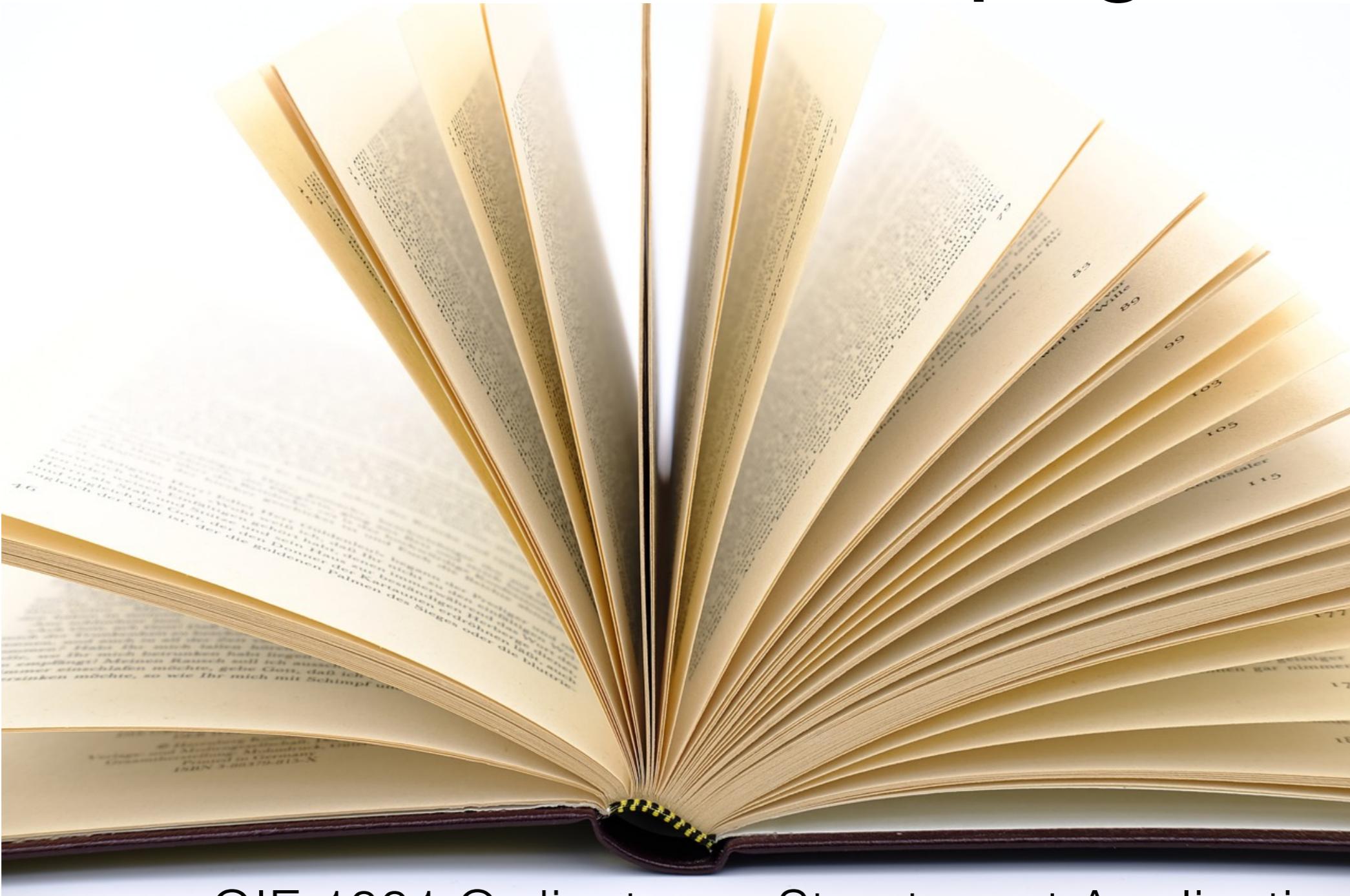


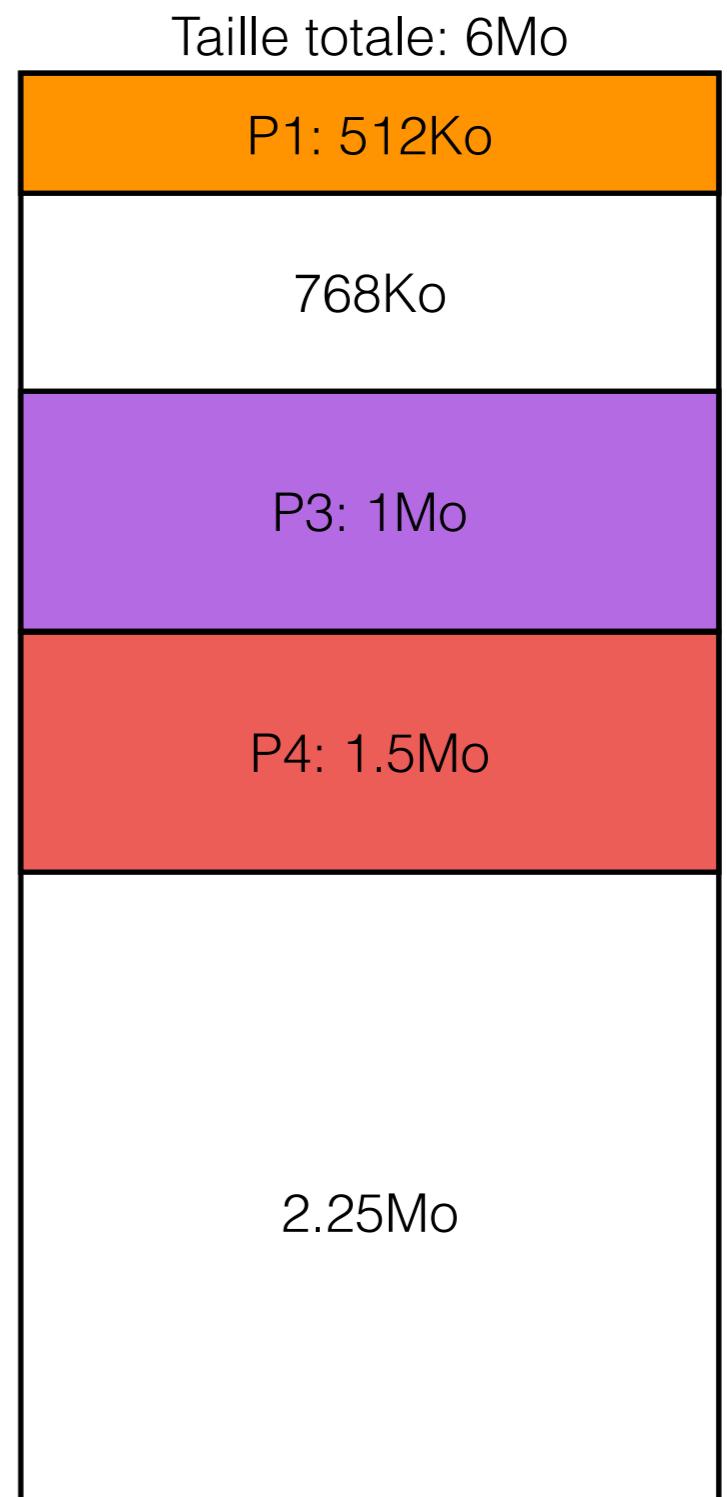
Allocation mémoire paginée



GIF-1001 Ordinateurs: Structure et Applications
Jean-François Lalonde

Question

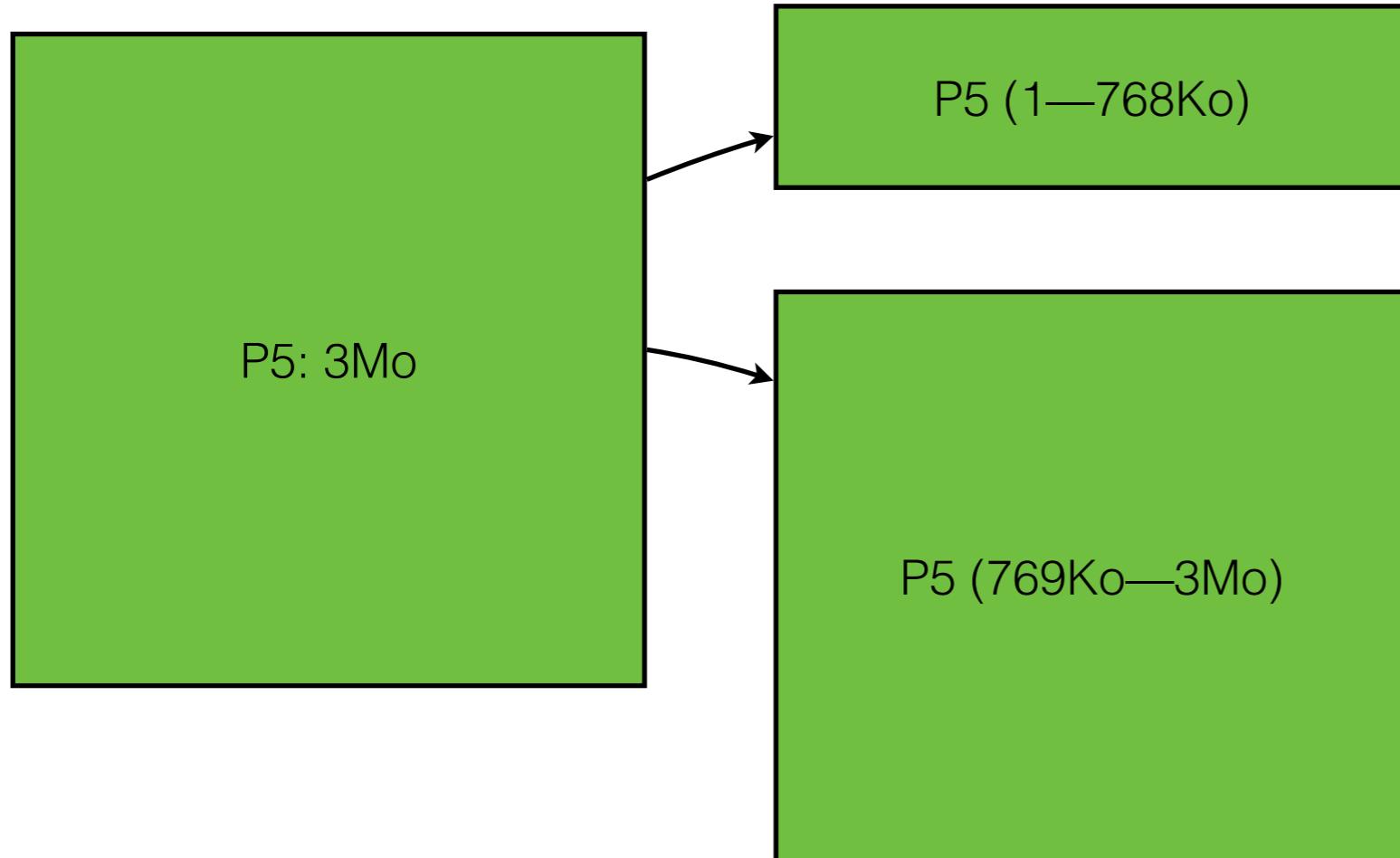
Que faire avec P5?



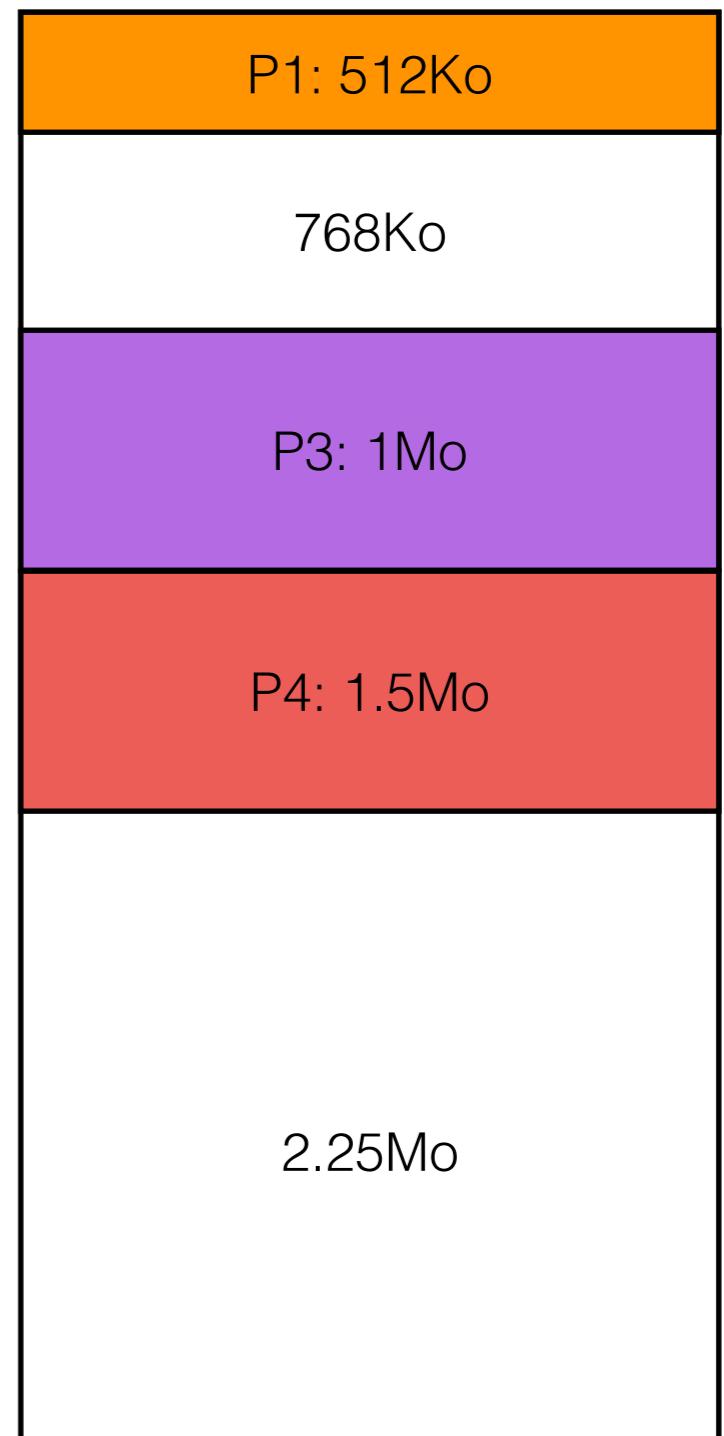
Question

Le séparer en deux blocs?

Que faire avec P5?



Taille totale: 6Mo



Question

Que faire avec P5?

Le séparer en deux blocs?

Placer chacun des blocs dans les espaces libres?

Taille totale: 6Mo



Cette stratégie
(subdiviser un processus en morceaux de **différentes** tailles)
rend le MMU très complexe...

P5: 3Mo

L'approche que nous verrons subdivise un processus en morceaux ayant tous la **même** taille,
ce qui simplifiera le MMU!

The diagram shows a large green box representing process P5 with a total size of 3Mo. It is divided into three horizontal sections of equal size: a small green section at the bottom labeled 'P5 (769Ko—3Mo)', a medium green section in the middle, and a large green section at the top. A green bar at the bottom left indicates the total size of 3Mo.

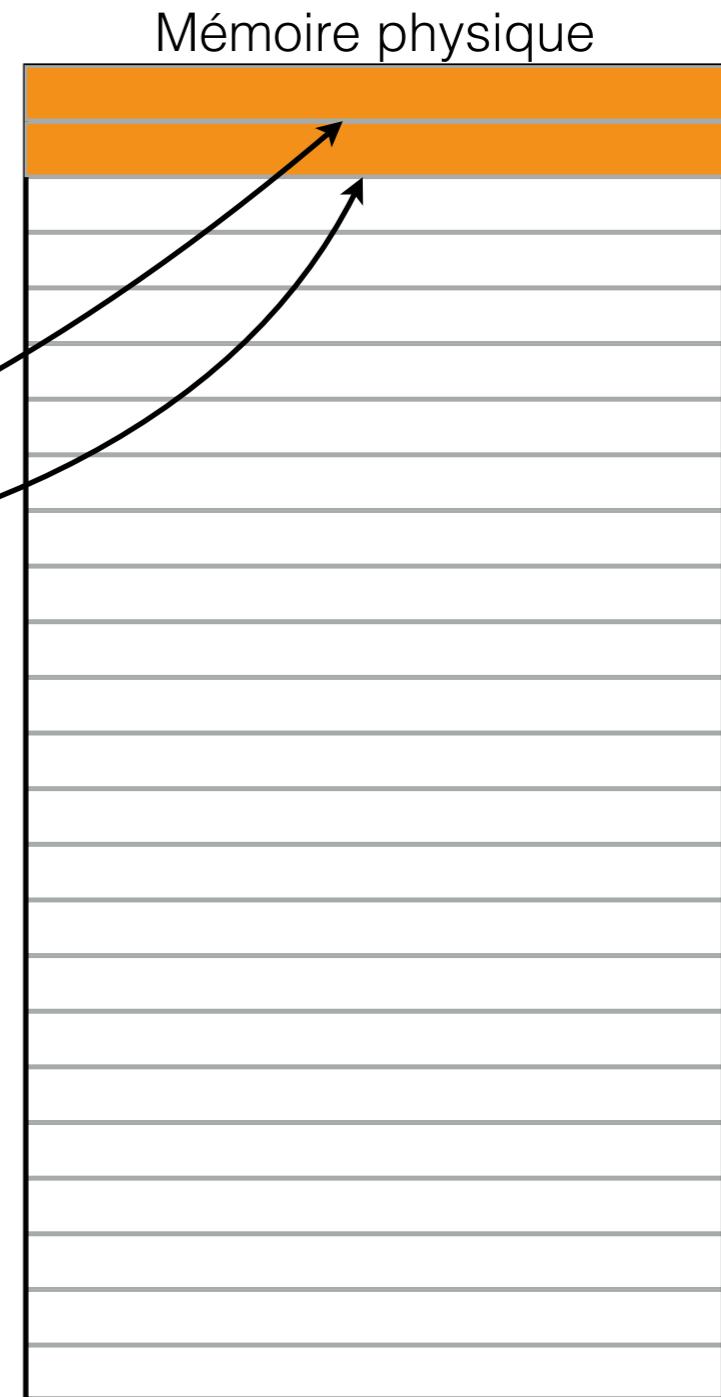
Allocation mémoire paginée

Allocation mémoire paginée

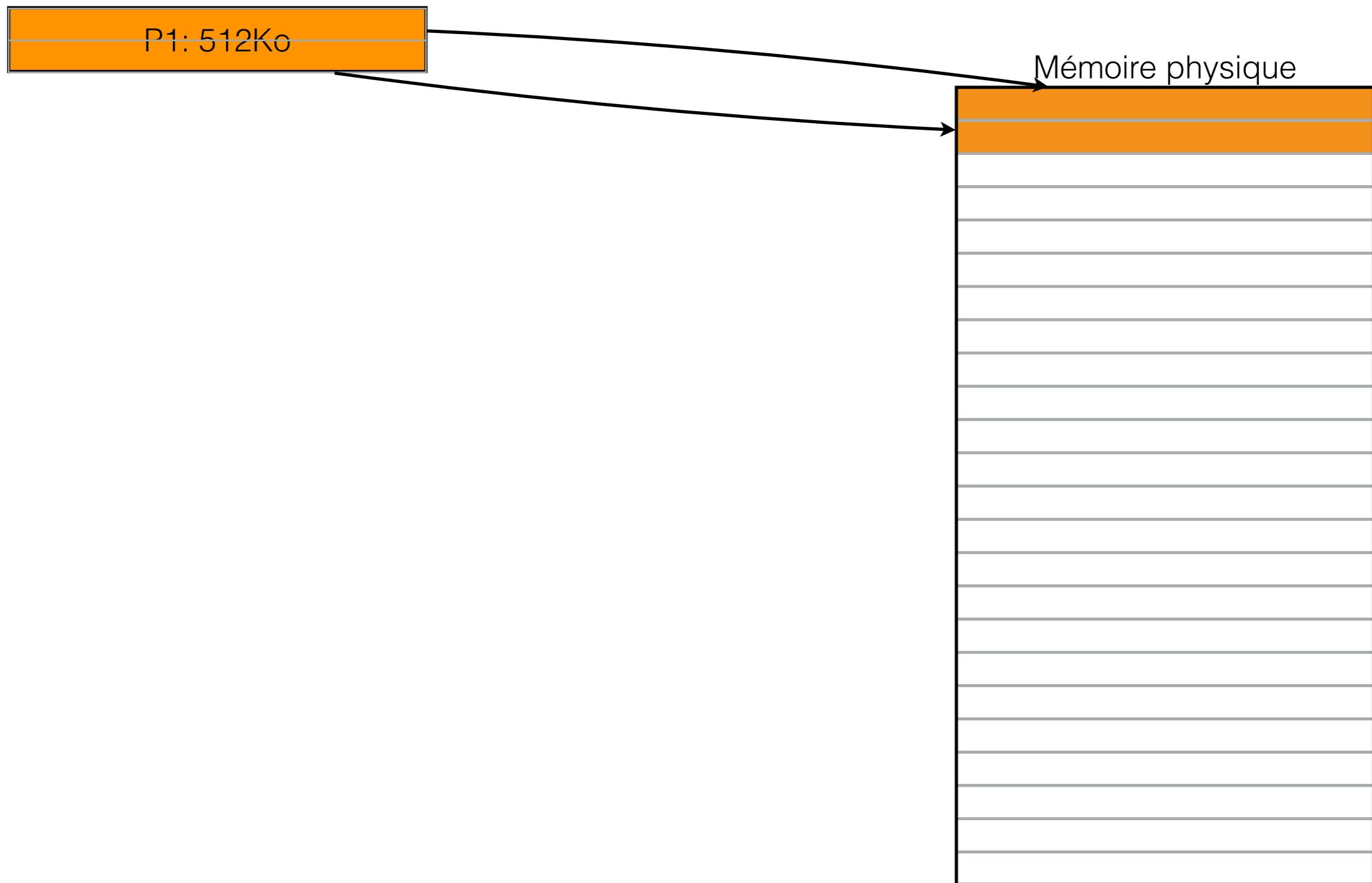
- On divise la mémoire physique en petits morceaux nommés **trames** (*frames*)
- On divise la mémoire virtuelle en petits morceaux nommés **pages**



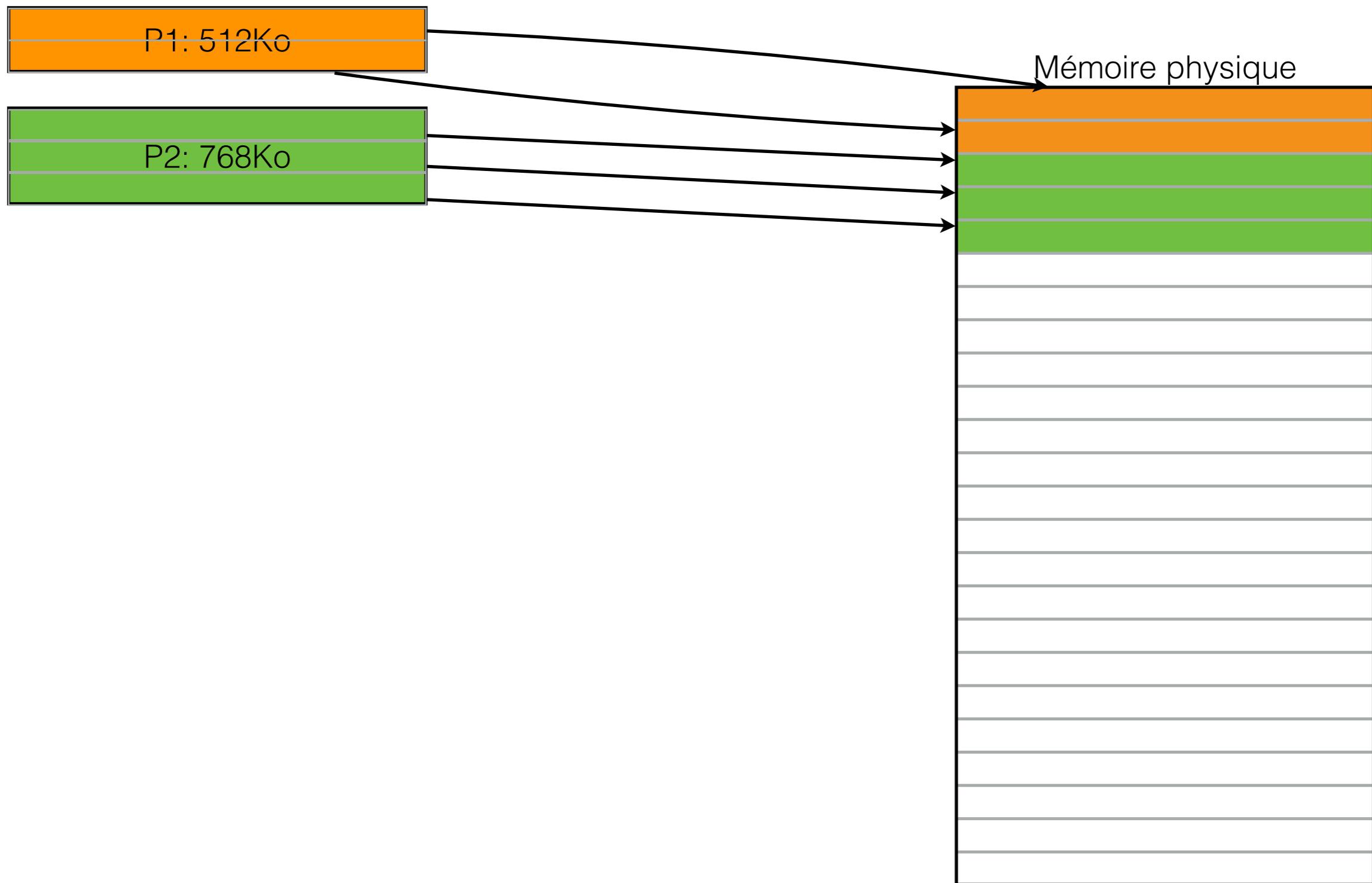
- Une trame a **toujours** la même taille qu'une page. Dans l'exemple, la taille d'une page/trame est de 256Ko
- Lorsqu'on alloue de la mémoire à un processus, on **attribue une page à une trame**



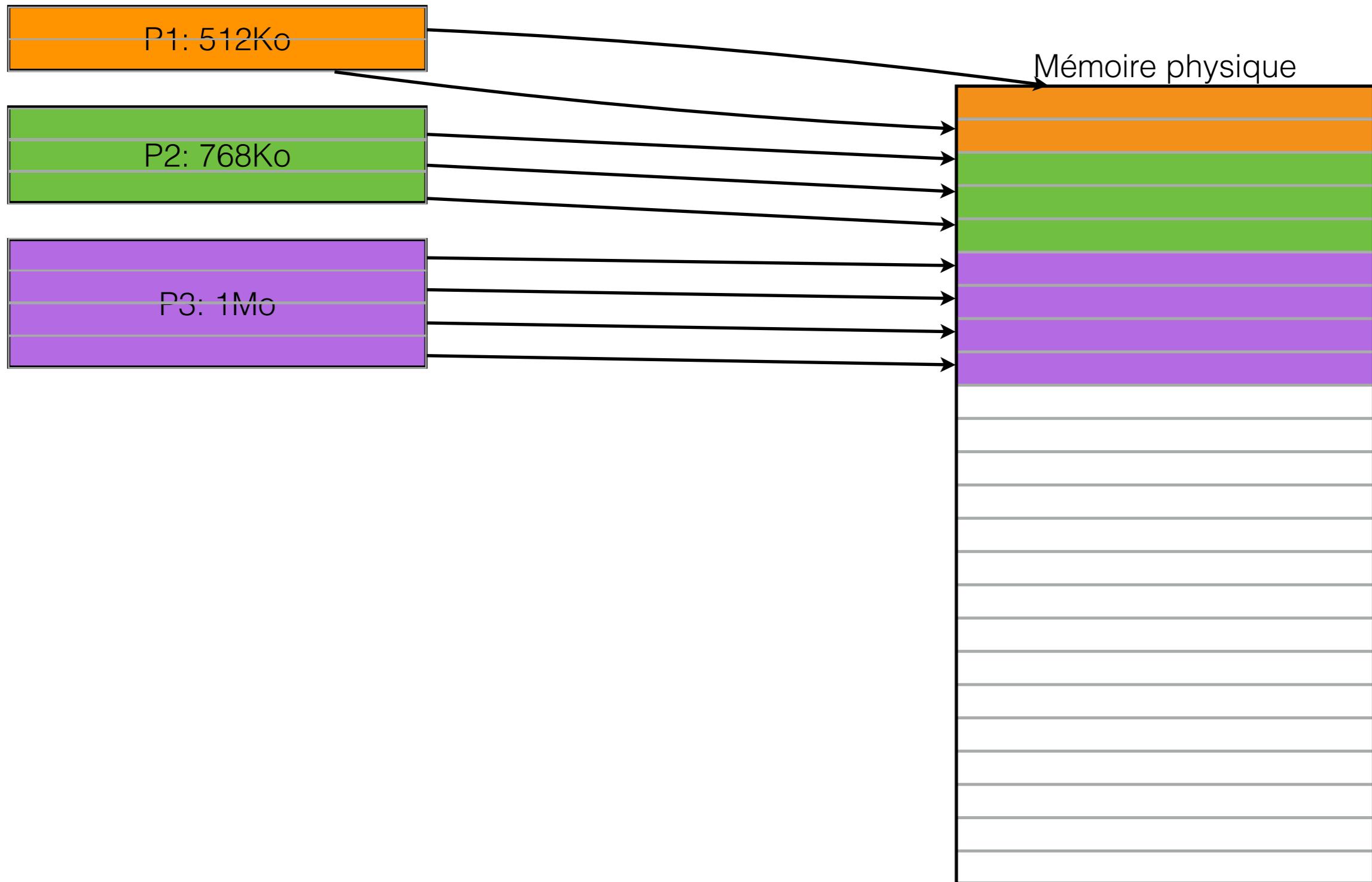
Allocation mémoire paginée



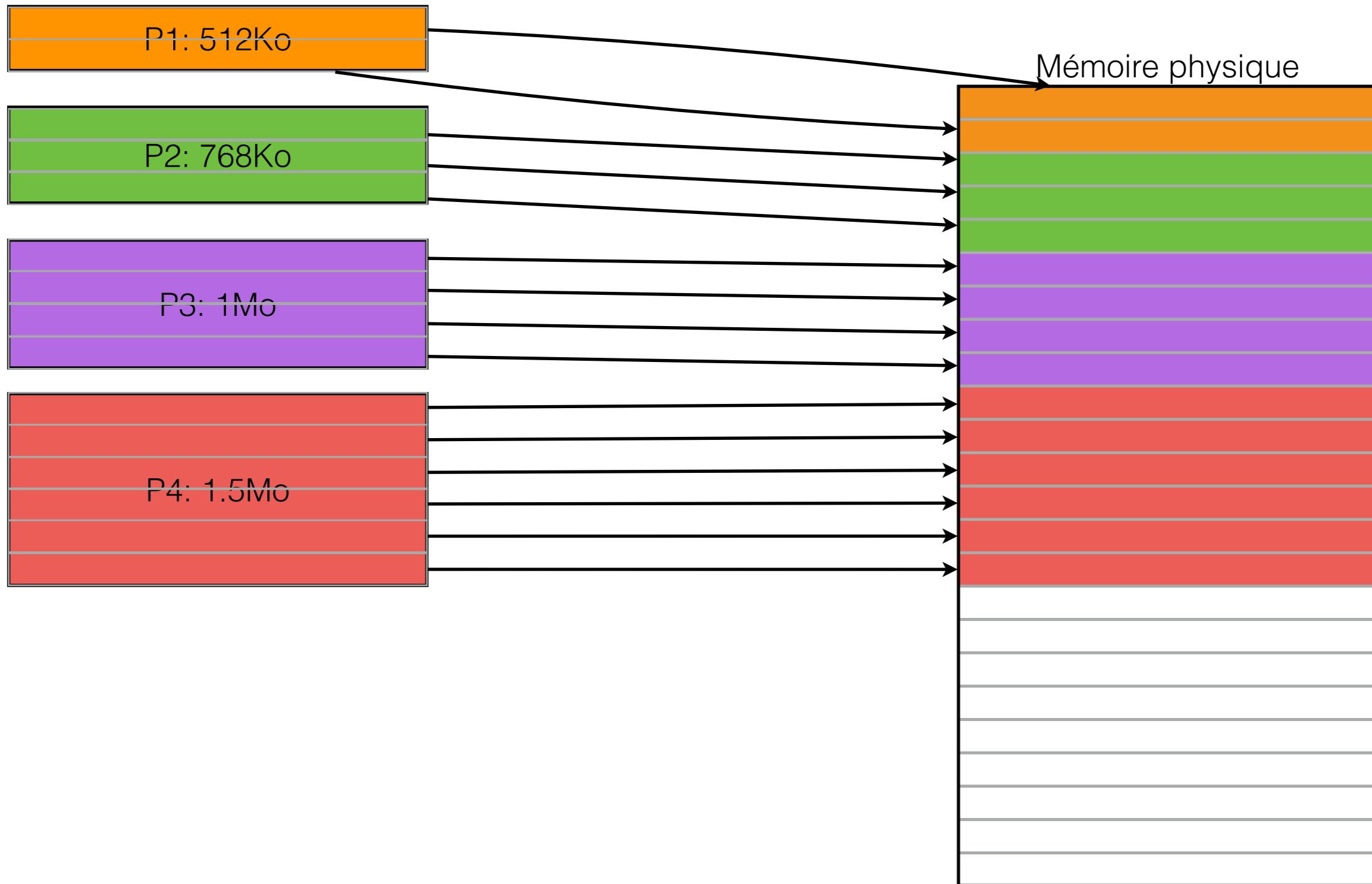
Allocation mémoire paginée



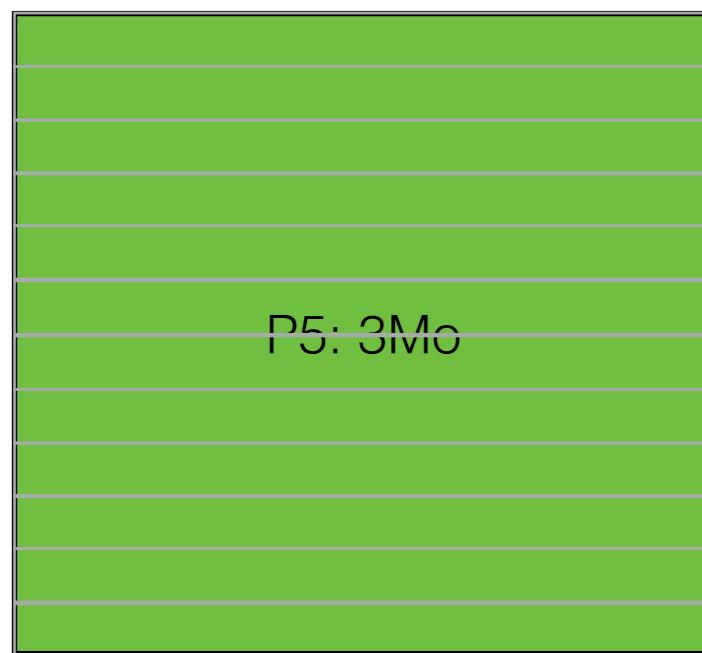
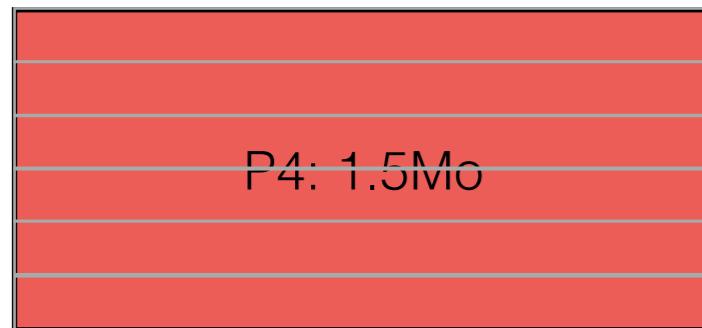
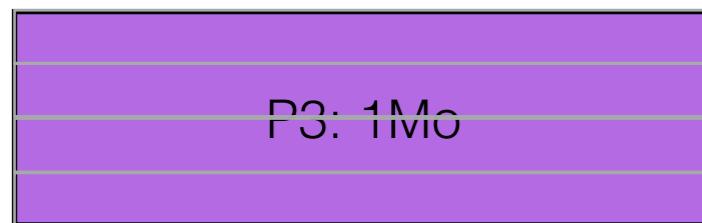
Allocation mémoire paginée



Allocation mémoire paginée



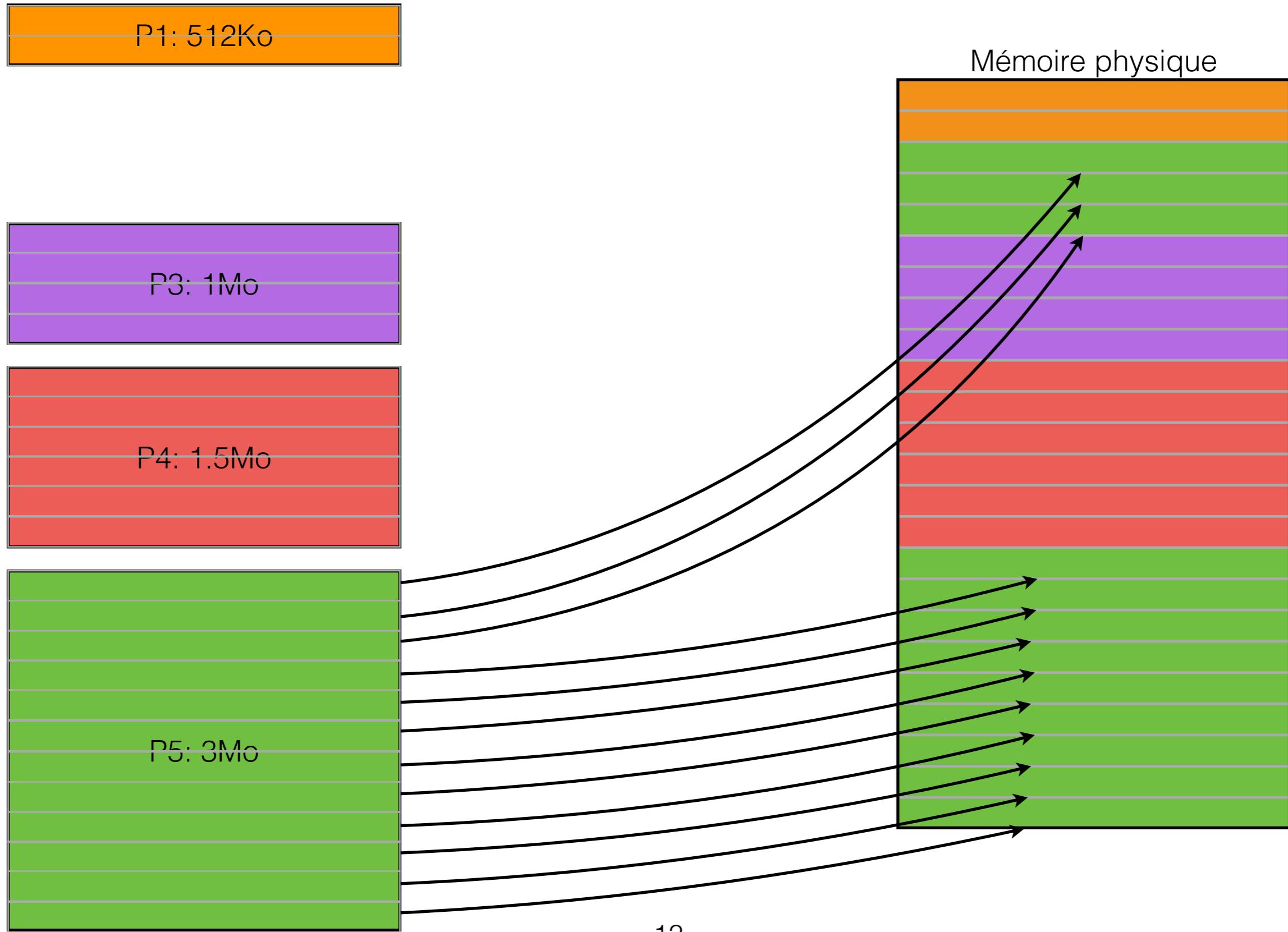
Allocation mémoire paginée



P2 se termine

P5 est admis

Allocation mémoire paginée

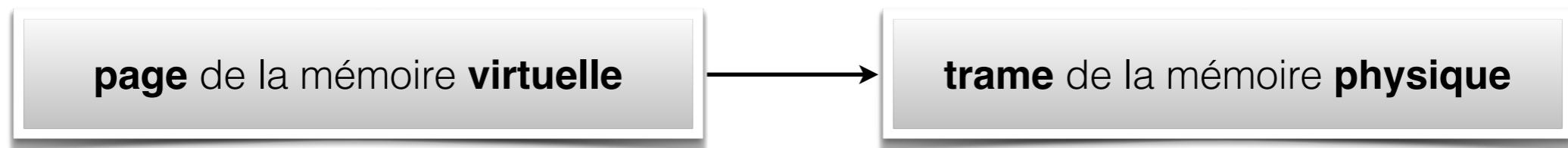


Questions

- Comment fait-on pour savoir
 - l'endroit où une page est stockée?
 - la correspondance entre une page et une trame?

La table des pages

- La **table des pages** est une structure de données dans le MMU qui conserve la correspondance entre chaque:



En pratique, on ne stocke pas le numéro de page en mémoire.
Le numéro de ligne dans la table correspond au numéro de page.



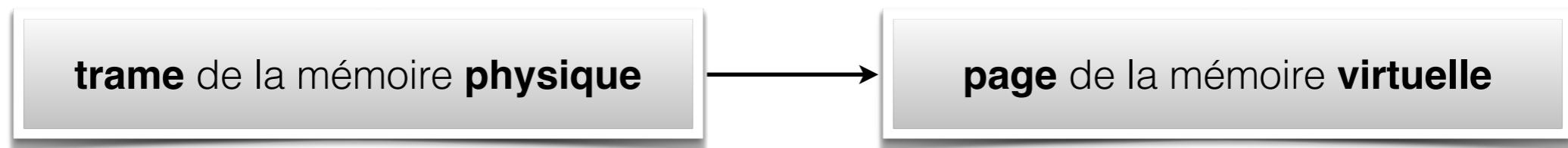
| # page | # trame |
|--------|---------|
| 0x0 | 0xAB3 |
| 0x1 | 0x3A2 |
| 0x2 | 0x123 |
| 0x3 | 0x2F3 |
| 0x4 | 0x124 |
| 0x5 | 0x2F0 |
| 0x6 | 0x125 |
| : | : |

Table des pages inversée

- Une table des pages indique:
 - le numéro de **trame** correspondant à chaque **page**.
- Il peut aussi être utile de savoir l'inverse:
 - le numéro de **page** correspondant à chaque **trame**.
- C'est la **table des pages inversée**.
- La taille d'une table des pages inversée est proportionnelle à la taille de la mémoire **physique**.

La table des pages inversées

- La **table des pages inversées** est une structure de données dans le MMU qui conserve la correspondance entre chaque:



C'est un peu comme une carte indiquant le contenu de chaque trame en mémoire physique (RAM).



| # trame | # page |
|---------|--------|
| : | : |
| 0x122 | 0x3A2 |
| 0x123 | 0x2 |
| 0x124 | 0x4 |
| 0x125 | 0x6 |
| 0x126 | 0x2F0 |
| 0x127 | 0x10F |
| : | : |

Taille de la table des pages

- Le nombre de lignes: il y a **une ligne par page**
- L'information stockée dans chaque ligne: le **numéro de trame correspondant à chaque page**
 - combien de bits sont nécessaires pour représenter le numéro de trame?
 - $\log_2(\text{nb trames})$
- Calcul de la taille
 - taille = (nb pages) \times (nb bits nécessaires pour représenter une trame)
- *Rappel:* il n'est pas nécessaire de stocker le numéro de page, nous pouvons utiliser le numéro de ligne correspondant à la page.
 - par exemple, la page 0x2D (45) est à la ligne 45 dans la table
 - la première ligne correspond à la page 0

Table des pages: équations

Taille de la table des pages:

$$\begin{aligned}\text{taille} &= (\text{nb pages}) \times (\text{nb bits nécessaires pour représenter une trame}) \\ &= (\text{nb pages}) \times \log_2(\text{nb trames})\end{aligned}$$

Nombre de pages et de trames:

$$\text{nb pages} = \frac{\text{taille mémoire virtuelle}}{\text{taille d'une page}} \quad \text{nb trames} = \frac{\text{taille mémoire physique}}{\text{taille d'une trame}}$$

Taille de la table des pages

- Un système possède les caractéristiques suivantes:
 - une mémoire *physique* de 32Mo
 - une mémoire *virtuelle* de 64Mo
 - la taille d'une page est de 4Ko
- Quelle est la taille de la table des pages?

Taille de la table des pages

- Un système possède les caractéristiques suivantes:
 - une mémoire *physique* de 32Mo
 - une mémoire *virtuelle* de 64Mo
 - la taille d'une page est de 4Ko
- Quelle est la taille de la table des pages?

Rappel

$$\text{taille} = (\text{nb pages}) \times \log_2(\text{nb trames})$$

$$\text{nb pages} = \frac{\text{taille mémoire virtuelle}}{\text{taille d'une page}}$$

$$\text{nb trames} = \frac{\text{taille mémoire physique}}{\text{taille d'une trame}}$$

Taille de la table des pages

- Un système possède les caractéristiques suivantes:
 - une mémoire *physique* de 32Mo
 - une mémoire *virtuelle* de 64Mo
 - la taille d'une page est de 4Ko
- Quelle est la taille de la table des pages?

Rappel

$$\text{taille} = (\text{nb pages}) \times \log_2(\text{nb trames})$$

$$\text{nb pages} = \frac{\text{taille mémoire virtuelle}}{\text{taille d'une page}}$$

$$\text{nb trames} = \frac{\text{taille mémoire physique}}{\text{taille d'une trame}}$$

- Déterminer le nombre de *pages*

$$64\text{Mo} / 4\text{Ko} = 64 \times 2^{20} / 4 \times 2^{10} = 2^{26} / 2^{12} = 2^{14} \text{ pages}$$

- Déterminer le nombre de *trames*

$$32\text{Mo} / 4\text{Ko} = 32 \times 2^{20} / 4 \times 2^{10} = 2^{25} / 2^{12} = 2^{13} \text{ trames}$$

- Déterminer le nombre de bits nécessaires pour stocker le # de trame

$$\log_2(2^{13}) = 13 \text{ bits}$$

- Calculer la taille totale (#pages x #bits)

$$2^{14} \times 13 = 2^4 \times 13 \times 2^{10} = 208 \times 2^{10} = 208\text{Kbits} = 26\text{Ko}$$

Allocation paginée: traduction d'adresses

- Dans le cas de l'allocation mémoire paginée, le MMU est plus complexe que pour l'allocation contigüe
 - On ne peut plus simplement additionner la première adresse!
- Pour traduire l'adresse, il faut effectuer trois étapes:
 1. Déterminer la **page** de l'adresse **virtuelle**;
 2. Trouver la **trame** (dans la mémoire **physique**) correspondant à cette page;
 3. Remplacer le numéro de page par le numéro de trame.

1. Déterminer la **page** de l'adresse **virtuelle**

- Une adresse virtuelle est divisée en deux:
 - le numéro de page (MSB)
 - la position dans la page (LSB)
- Le nombre de bits utilisé pour ces deux parties dépend de la **taille d'une page**
- Exemple en décimal:
 - Une page contient 100 mots
 - Combien de symboles (chiffres) ai-je besoin pour identifier le mot dans la page?
nb symboles = $\log_{10}(100) = 2$, soit de 00 à 99
 - Dans quelle page est situé le mot 425?

| Page | Position |
|------|----------|
| 4 | 2 5 |

1. Déterminer la **page** de l'adresse **virtuelle**

- Une adresse virtuelle est divisée en deux:
 - le numéro de page (MSB)
 - la position dans la page (LSB)
- Le nombre de bits utilisé pour ces deux parties dépend de la **taille d'une page**
- Exemple en binaire:
 - Une page contient 4Ko. Chaque octet possède une adresse, donc, 4K adresses.
 - Combien de symboles (bits) ai-je besoin pour identifier l'adresse dans la page?
$$\text{nb bits} = \log_2(4 \times 2^{10}) = \log_2(2^{12}) = 12 \text{ bits}$$
 - Dans quelle page est situé l'adresse 0x3D7A1?

| Page | Position |
|--------|----------|
| 0x 3 D | 7 A 1 |

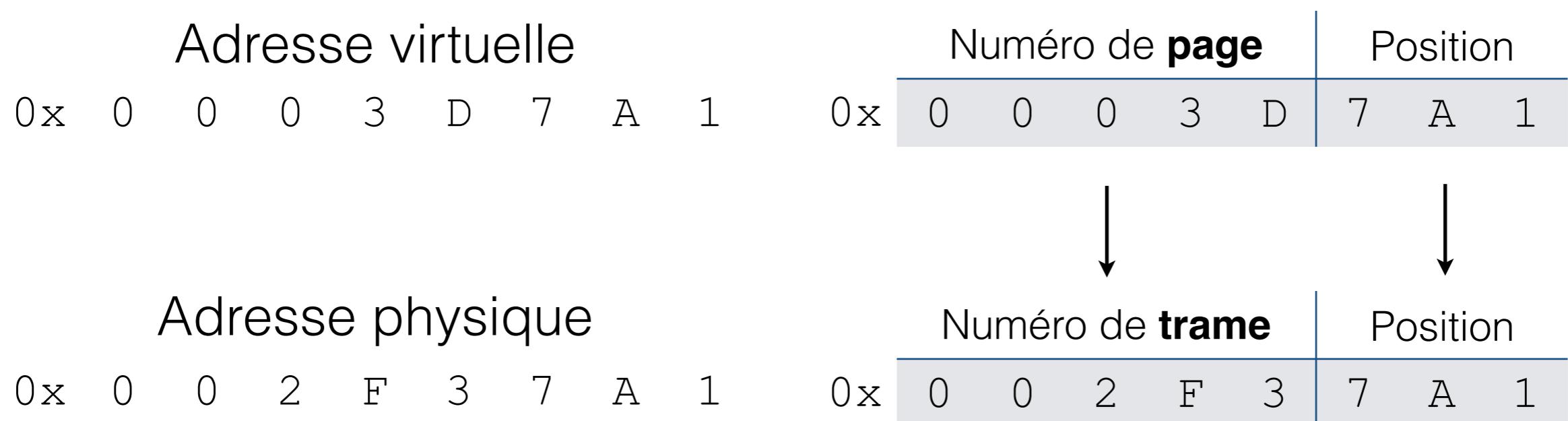
2. Trouver la **trame** correspondant à cette **page**

- Comment fait-on pour savoir
 - l'endroit où une page est stockée?
 - la correspondance entre une page et une trame?
- On utilise la **table des pages**, une structure de données dans le MMU qui conserve la correspondance entre chaque:
 - **page** de la mémoire **virtuelle**
 - **trame** de la mémoire **physique**

On y reviendra dans une prochaine capsule!

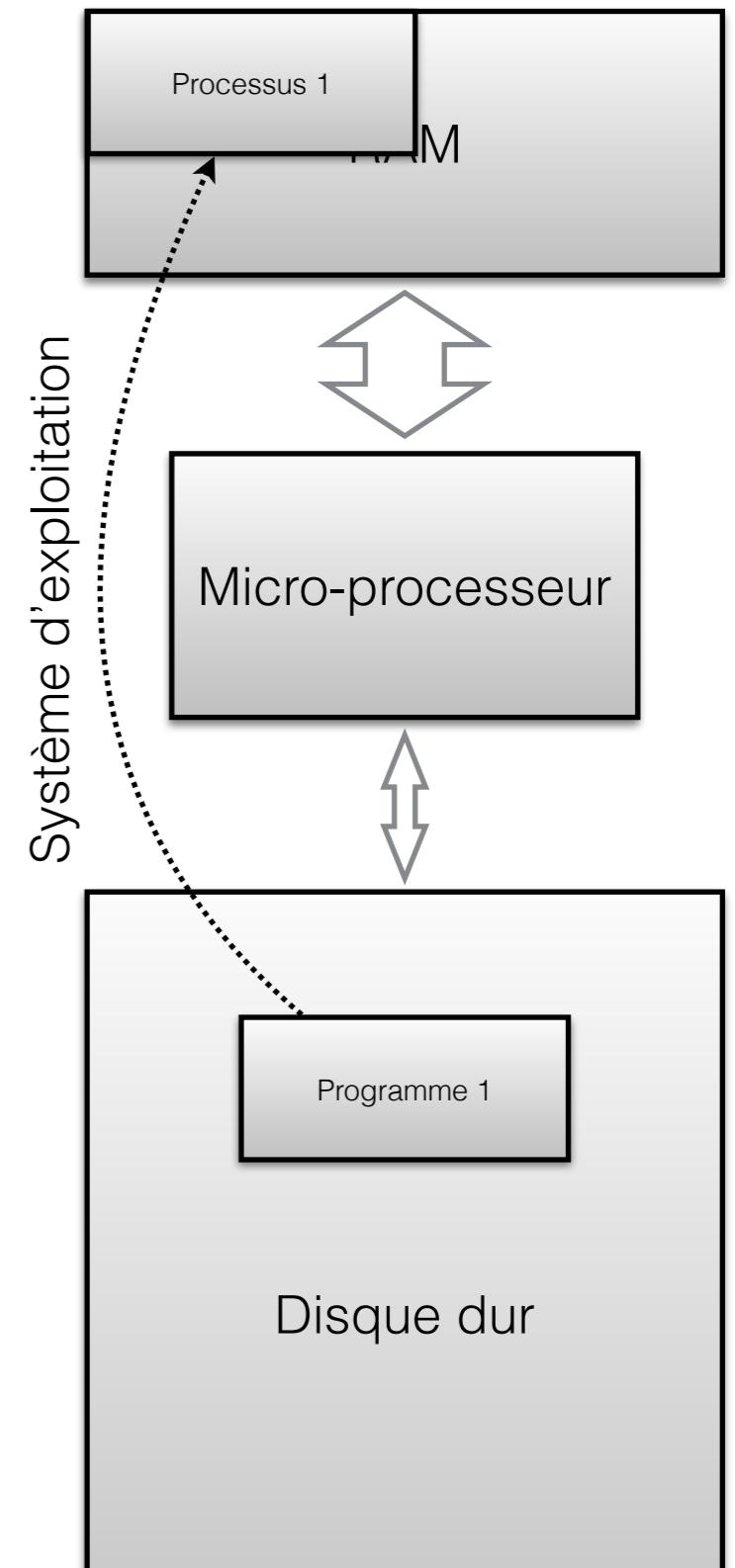
3. Remplacer le # de **page** par le # de **trame**

- Supposons que la table des pages révèle que la page 0x3D correspond à la trame 0x2F3
- Il suffit de remplacer le numéro de page par le numéro de trame pour obtenir l'adresse **physique**

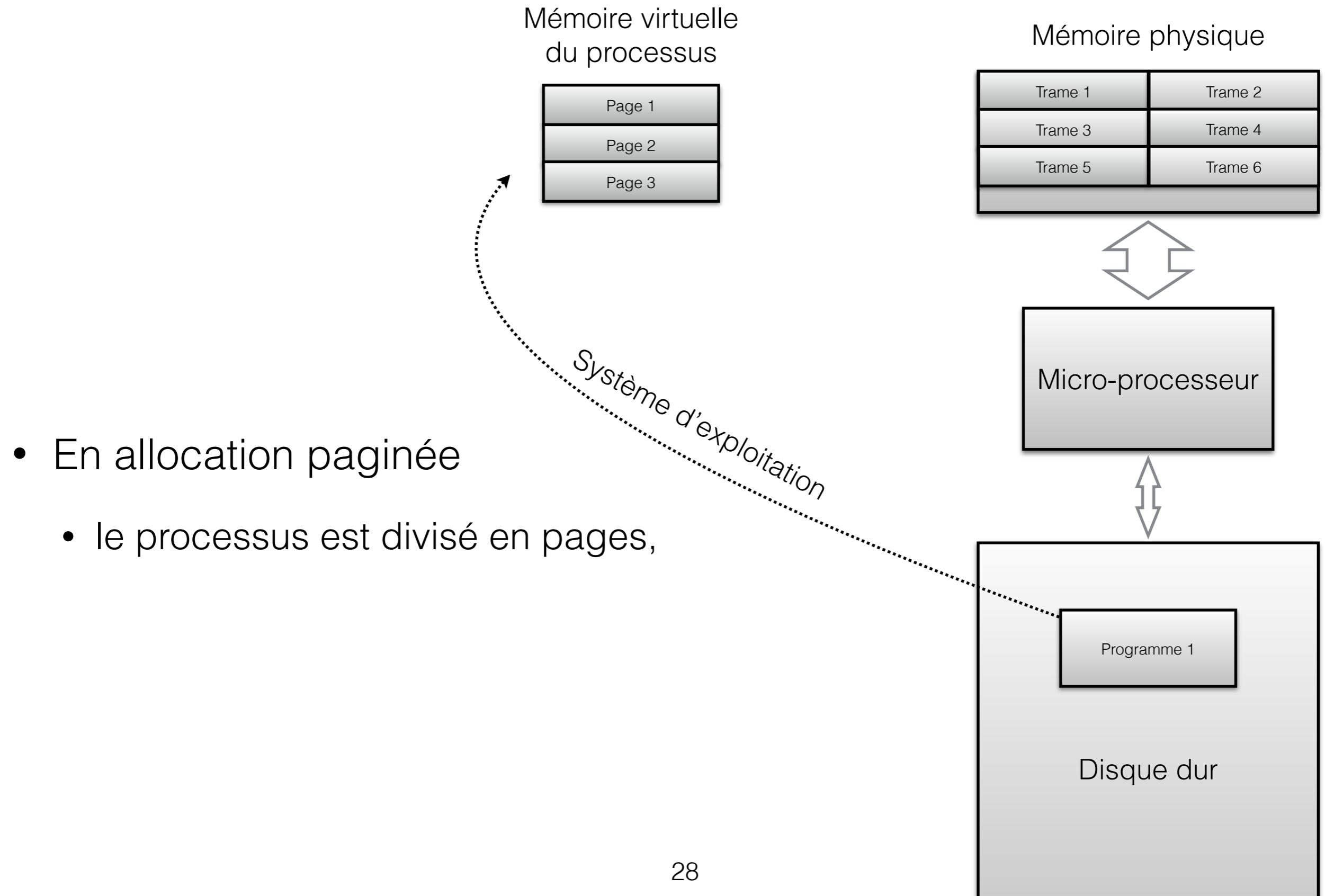


Création d'un processus

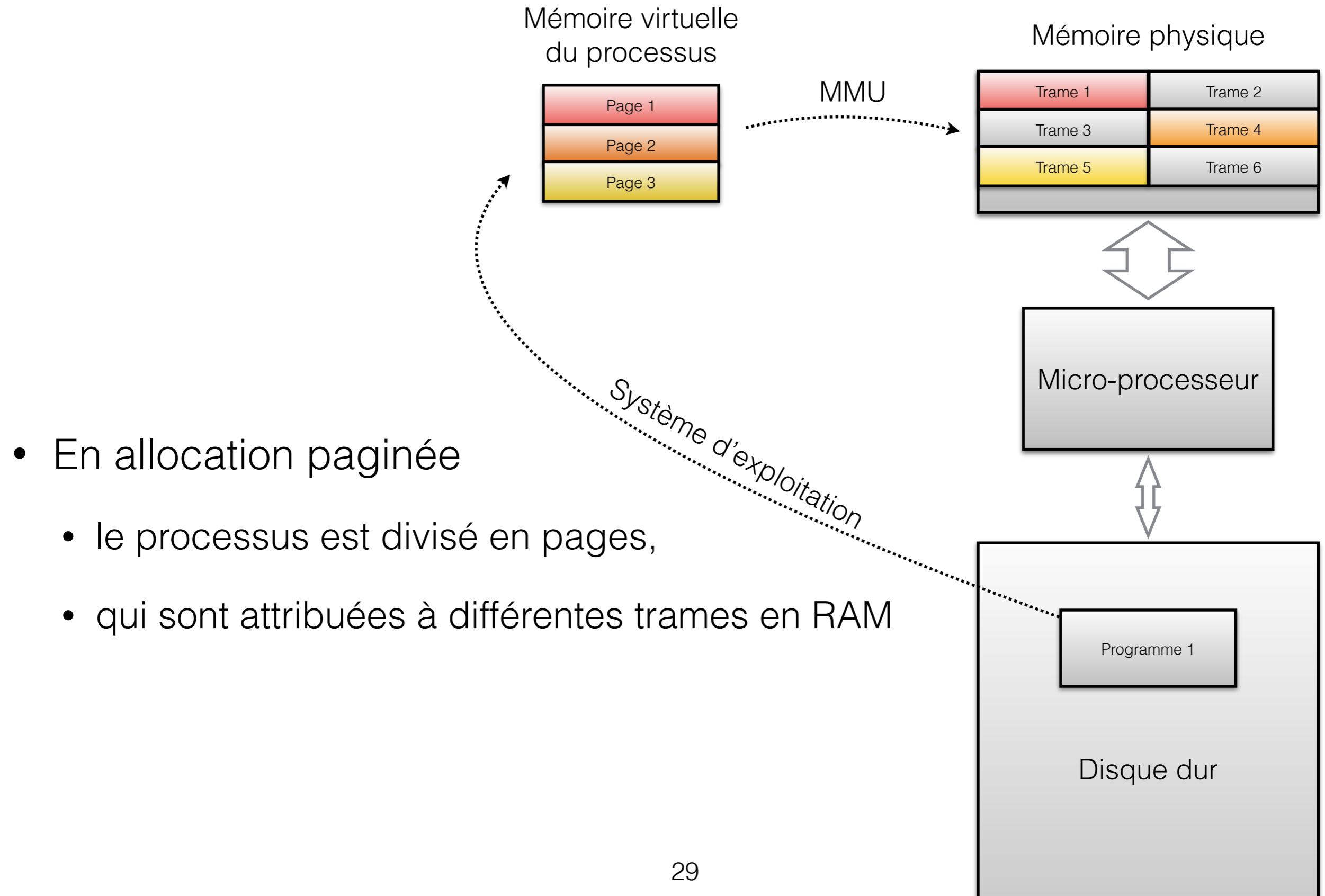
- Un ordinateur possède plusieurs programmes, situés sur le disque dur
- Lors du démarrage du programme
 - Les ressources nécessaires (mémoire, etc.) sont créées par le système d'exploitation lorsque le programme est démarré



Création d'un processus – allocation paginée



Création d'un processus – allocation paginée



Création d'un processus – allocation paginée

Question

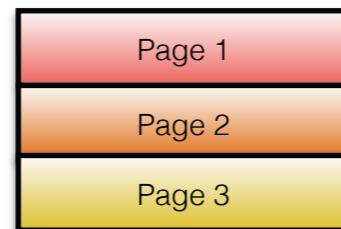
Est-ce qu'un processus doit être entièrement en mémoire?

Non!

Seules les pages nécessaires à l'exécution doivent être en mémoire.

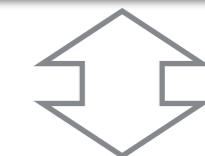
- En allocation paginée
 - le processus est divisé en pages,
 - qui sont attribuées à différentes trames en RAM

Mémoire virtuelle
du processus



MMU

Mémoire physique



Micro-processeur

Système d'exploitation

Programme 1

Disque dur

FAQ

Laquelle est plus grosse:
la mémoire **virtuelle** (pour tous les processus)
ou la mémoire **physique**?

Où est-ce que les deux doivent avoir la même taille?

La mémoire **virtuelle** peut être beaucoup plus grosse
que la mémoire **physique**!

Conditions

- 2 conditions pour qu'un programme puisse s'exécuter:
 - l'instruction (ou la donnée) nécessaire doit être en mémoire RAM
 - la table de pages pour ce programme doit contenir une entrée qui traduit l'adresse (virtuelle) du programme vers l'adresse (physique) en RAM

Allocation dynamique de pages

- Pour un système possédant les caractéristiques suivantes:
 - Pages de 4Ko
 - Mémoire physique (RAM) de 16Ko
 - Mémoire virtuelle de 32Ko
 - Les pages suivantes sont en mémoire:
 - page 0 en trame 1
 - page 1 en trame 2
 - page 3 en trame 0
1. Illustrez la table des pages et la table des pages inversées
 2. Effectuez la séquence d'opérations suivantes en mettant à jour les tables. Indiquez les adresses physiques correspondantes.
 1. Charger valeur (e.g. LDR) à l'adresse 0x0A38
 2. Écrire une valeur (e.g. STR) à l'adresse 0x3B97
 3. Charger la valeur (e.g. LDR) à l'adresse 0x2928
 4. Brancher (e.g. B) à l'adresse 0x7BF0

Allocation dynamique de pages

- Pour un système possédant les caractéristiques suivantes:
 - Pages de 4Ko
 - Mémoire physique (RAM) de 16Ko
 - Mémoire virtuelle de 32Ko
- Les pages suivantes sont en mémoire:
 - page 0 en trame 1
 - page 1 en trame 2
 - page 3 en trame 0

1. Illustrez la table des pages et la table des pages inversées

2. Effectuez la séquence d'opérations suivantes en mettant à jour les tables. Indiquez les adresses physiques correspondantes.

1. Charger valeur (e.g. LDR) à l'adresse 0x0A38
2. Écrire une valeur (e.g. STR) à l'adresse 0x3B97
3. Charger la valeur (e.g. LDR) à l'adresse 0x2928
4. Brancher (e.g. B) à l'adresse 0x7BF0

Allocation dynamique de pages

- Pour un système possédant les caractéristiques suivantes:
 - Pages de 4Ko
 - Mémoire physique (RAM) de 16Ko
 - Mémoire virtuelle de 32Ko
- Les pages suivantes sont en mémoire:
 - page 0 en trame 1
 - page 1 en trame 2
 - page 3 en trame 0

1. Illustrez la table des pages et la table des pages inversées

2. Effectuez la séquence d'opérations suivantes en mettant à jour les tables. Indiquez les adresses physiques correspondantes.
 1. Charger valeur (e.g. LDR) à l'adresse 0x0A38
 2. Écrire une valeur (e.g. STR) à l'adresse 0x3B97
 3. Charger la valeur (e.g. LDR) à l'adresse 0x2928
 4. Brancher (e.g. B) à l'adresse 0x7BF0

Table des pages

| # page | # trame |
|--------|---------|
| 0 | |
| 1 | |
| 2 | |
| 3 | |
| 4 | |
| 5 | |
| 6 | |
| 7 | |

Table des pages inversée

| # trame | # page |
|---------|--------|
| 0 | |
| 1 | |
| 2 | |
| 3 | |

Allocation dynamique de pages

Table des pages

- Pour un système possédant les caractéristiques suivantes:

- Pages de 4Ko
- Mémoire physique (RAM) de 16Ko
- Mémoire virtuelle de 32Ko

- Les pages suivantes sont en mémoire:
 - page 0 en trame 1
 - page 1 en trame 2
 - page 3 en trame 0

1. Illustrez la table des pages et la table des pages inversées

2. Effectuez la séquence d'opérations suivantes en mettant à jour les tables. Indiquez les adresses physiques correspondantes.

- Charger valeur (e.g. LDR) à l'adresse 0x0A38
- Écrire une valeur (e.g. STR) à l'adresse 0x3B97
- Charger la valeur (e.g. LDR) à l'adresse 0x2928
- Brancher (e.g. B) à l'adresse 0x7BF0

Table des pages:

$$\frac{\text{taille mémoire virtuelle}}{\text{taille d'une page}} = \frac{32\text{Ko}}{4\text{Ko}} = 8$$

Table des pages inversées:

$$\frac{\text{taille mémoire physique}}{\text{taille d'une trame}} = \frac{16\text{Ko}}{4\text{Ko}} = 4$$

| # page | # trame |
|--------|---------|
| 0 | 1 |
| 1 | 2 |
| 2 | |
| 3 | 0 |
| 4 | |
| 5 | |
| 6 | |
| 7 | |

Table des pages inversée

| # trame | # page |
|---------|--------|
| 0 | 3 |
| 1 | 0 |
| 2 | 1 |
| 3 | |

Allocation dynamique de pages

2. Effectuez la séquence d'opérations suivantes en mettant à jour les tables. Indiquez les adresses physiques correspondantes.

1. Charger valeur (e.g. LDR) à l'adresse 0x0A38

2. Écrire une valeur (e.g. STR) à l'adresse 0x3B97
3. Charger la valeur (e.g. LDR) à l'adresse 0x2928
4. Brancher (e.g. B) à l'adresse 0x7BF0

Table des pages

| # page | # trame |
|--------|---------|
| 0 | 1 |
| 1 | 2 |
| 2 | |
| 3 | 0 |
| 4 | |
| 5 | |
| 6 | |
| 7 | |

Table des pages inversée

| # trame | # page |
|---------|--------|
| 0 | 3 |
| 1 | 0 |
| 2 | 1 |
| 3 | |

Allocation dynamique de pages

2. Effectuez la séquence d'opérations suivantes en mettant à jour les tables. Indiquez les adresses physiques correspondantes.

1. Charger valeur (e.g. LDR) à l'adresse 0x0A38

2. Écrire une valeur (e.g. STR) à l'adresse 0x3B97

3. Charger la valeur (e.g. LDR) à l'adresse 0x2928

4. Brancher (e.g. B) à l'adresse 0x7BF0

Nombre de bits pour la position dans la page:

$$\log_2(\text{taille d'une page}) = \log_2(4 \times 2^{10}) = \log_2(2^{12}) = 12 \text{ bits}$$

| adresse virtuelle: | 0x | 0 | Numéro | Position | | | 4 |
|--------------------|----|---|---------|---------------|---|---|---|
| | | | de page | A | 3 | 8 | |
| adresse physique: | 0x | 1 | Numéro | Position | | | 5 |
| | | | trame | A | 3 | 8 | 6 |
| | | | | Position | | | 7 |
| | | | | dans la trame | | | |

Table des pages

| # page | # trame |
|--------|---------|
| 0 | 1 |
| 1 | 2 |
| 2 | |
| 3 | 0 |
| 4 | |
| 5 | |
| 6 | |
| 7 | |

Table des pages inversée

| # trame | # page |
|---------|--------|
| 0 | 3 |
| 1 | 0 |
| 2 | 1 |
| 3 | |

Allocation dynamique de pages

2. Effectuez la séquence d'opérations suivantes en mettant à jour les tables. Indiquez les adresses physiques correspondantes.

1. Charger valeur (e.g. LDR) à l'adresse 0x0A38
- 2. Écrire une valeur (e.g. STR) à l'adresse 0x3B97**
3. Charger la valeur (e.g. LDR) à l'adresse 0x2928
4. Brancher (e.g. B) à l'adresse 0x7BF0

Table des pages

| # page | # trame |
|--------|---------|
| 0 | 1 |
| 1 | 2 |
| 2 | |
| 3 | 0 |
| 4 | |
| 5 | |
| 6 | |
| 7 | |

Table des pages inversée

| # trame | # page |
|---------|--------|
| 0 | 3 |
| 1 | 0 |
| 2 | 1 |
| 3 | |

Allocation dynamique de pages

2. Effectuez la séquence d'opérations suivantes en mettant à jour les tables. Indiquez les adresses physiques correspondantes.

1. Charger valeur (e.g. LDR) à l'adresse 0x0A38

2. Écrire une valeur (e.g. STR) à l'adresse 0x3B97

3. Charger la valeur (e.g. LDR) à l'adresse 0x2928

4. Brancher (e.g. B) à l'adresse 0x7BF0

Nombre de bits pour la position dans la page:

$$\log_2(\text{taille d'une page}) = \log_2(4 \times 2^{10}) = \log_2(2^{12}) = 12 \text{ bits}$$

| adresse virtuelle: | 0x | 3 | Numéro | Position | | | 4 |
|--------------------|----|---|---------|---------------|---|---|---|
| | | | de page | B | 9 | 7 | |
| adresse physique: | 0x | 0 | Numéro | Position | | | 5 |
| | | | trame | B | 9 | 7 | 6 |
| | | | | Position | | | 7 |
| | | | | dans la trame | | | |

Table des pages

| # page | # trame |
|--------|---------|
| 0 | 1 |
| 1 | 2 |
| 2 | |
| 3 | 0 |
| 4 | |
| 5 | |
| 6 | |
| 7 | |

Table des pages inversée

| # trame | # page |
|---------|--------|
| 0 | 3 |
| 1 | 0 |
| 2 | 1 |
| 3 | |

Allocation dynamique de pages

2. Effectuez la séquence d'opérations suivantes en mettant à jour les tables. Indiquez les adresses physiques correspondantes.

1. Charger valeur (e.g. LDR) à l'adresse 0x0A38
2. Écrire une valeur (e.g. STR) à l'adresse 0x3B97
- 3. Charger la valeur (e.g. LDR) à l'adresse 0x2928**
4. Brancher (e.g. B) à l'adresse 0x7BF0

Table des pages

| # page | # trame |
|--------|---------|
| 0 | 1 |
| 1 | 2 |
| 2 | |
| 3 | 0 |
| 4 | |
| 5 | |
| 6 | |
| 7 | |

Table des pages inversée

| # trame | # page |
|---------|--------|
| 0 | 3 |
| 1 | 0 |
| 2 | 1 |
| 3 | |

Allocation dynamique de pages

2. Effectuez la séquence d'opérations suivantes en mettant à jour les tables. Indiquez les adresses physiques corr

1. Charger valeur (e.g. LDR) à l'adresse 0x0A38
2. Écrire une valeur (e.g. STR) à l'adresse 0x3B97
- 3. Charger la valeur (e.g. LDR) à l'adresse 0x290**
4. Brancher (e.g. B) à l'adresse 0x7BF0

La page n'est pas allouée en RAM.
Il y a **faute de page!**

| adresse virtuelle: | 0x | 2 | Numéro | Position | 4 | |
|--------------------|----|----|----------|---------------|---|--|
| | | | de page | dans la page | | |
| adresse physique: | | 9 | Numéro | 2 | 5 | |
| | | | | 8 | | |
| | | 10 | Position | dans la trame | 6 | |
| | | | | | | |
| | | 11 | Position | dans la trame | 7 | |
| | | | | | | |

Table des pages

| # page | # trame |
|--------|---------|
| 0 | 1 |
| 1 | 2 |
| 2 | |
| 3 | 0 |
| 4 | |
| 5 | |
| 6 | |
| 7 | |

Table des pages inversée

| # trame | # page |
|---------|--------|
| 0 | 3 |
| 1 | 0 |
| 2 | 1 |
| 3 | |

Fautes de page

- Une **faute de page** survient lorsque le processus requiert une page qui n'est pas en mémoire.
- S'il **y a de la place** dans la mémoire physique:
 1. Créer la nouvelle page;
 2. Placer la page en mémoire physique dans une trame vide;
 3. Mettre à jour la table des pages et la table des pages inversée.

Allocation dynamique de pages

2. Effectuez la séquence d'opérations suivantes en mettant à jour les tables. Indiquez les adresses physiques correspondantes.

1. Charger valeur (e.g. LDR) à l'adresse 0x0A38
2. Écrire une valeur (e.g. STR) à l'adresse 0x3B97
- 3. Charger la valeur (e.g. LDR) à l'adresse 0x2928**
4. Brancher (e.g. B) à l'adresse 0x7BF0

Table des pages

| # page | # trame |
|--------|---------|
| 0 | 1 |
| 1 | 2 |
| 2 | |
| 3 | 0 |
| 4 | |
| 5 | |
| 6 | |
| 7 | |

Table des pages inversée

| # trame | # page |
|---------|--------|
| 0 | 3 |
| 1 | 0 |
| 2 | 1 |
| 3 | |

Allocation dynamique de pages

2. Effectuez la séquence d'opérations suivantes en mettant à jour les tables. Indiquez les adresses physiques corr

1. Charger valeur (e.g. LDR) à l'adresse 0x0A38
2. Écrire une valeur (e.g. STR) à l'adresse 0x3B97
- 3. Charger la valeur (e.g. LDR) à l'adresse 0x290**
4. Brancher (e.g. B) à l'adresse 0x7BF0

On met à jour la table des pages

Table des pages

| # page | # trame |
|--------|---------|
| 0 | 1 |
| 1 | 2 |
| 2 | 3 |
| 3 | 0 |
| 4 | |
| 5 | |
| 6 | |
| 7 | |

Nombre de bits pour la position dans la page:

$$\log_2(\text{taille d'une page}) = \log_2(4 \times 2^{10}) = \log_2(2^{12}) = 12 \text{ bits}$$

| adresse virtuelle: | 0x | 2 | Numéro | Position | | | 4 |
|--------------------|----|--------|---------|---------------|---|---|---|
| | | | de page | 9 | 2 | 8 | |
| adresse physique: | 0x | 3 | 9 | 2 | 8 | | 5 |
| | | Numéro | trame | Position | | | 6 |
| | | trame | | dans la trame | | | |

La trame 3 est libre.
On met à jour la table des pages inversée.

Table des pages inversée

| # trame | # page |
|---------|--------|
| 0 | 3 |
| 1 | 0 |
| 2 | 1 |
| 3 | 2 |

Allocation dynamique de pages

2. Effectuez la séquence d'opérations suivantes en mettant à jour les tables. Indiquez les adresses physiques correspondantes.

1. Charger valeur (e.g. LDR) à l'adresse 0x0A38
2. Écrire une valeur (e.g. STR) à l'adresse 0x3B97
3. Charger la valeur (e.g. LDR) à l'adresse 0x2928
- 4. Brancher (e.g. B) à l'adresse 0x7BF0**

Table des pages

| # page | # trame |
|--------|---------|
| 0 | 1 |
| 1 | 2 |
| 2 | 3 |
| 3 | 0 |
| 4 | |
| 5 | |
| 6 | |
| 7 | |

Table des pages inversée

| # trame | # page |
|---------|--------|
| 0 | 3 |
| 1 | 0 |
| 2 | 1 |
| 3 | 2 |

Allocation dynamique de pages

2. Effectuez la séquence d'opérations suivantes en mettant à jour les tables. Indiquez les adresses physiques corr

1. Charger valeur (e.g. LDR) à l'adresse 0x0A38
2. Écrire une valeur (e.g. STR) à l'adresse 0x3B97
3. Charger la valeur (e.g. LDR) à l'adresse 0x2920
4. Brancher (e.g. B) à l'adresse 0x7BF0

La page n'est pas allouée en RAM.
Il y a **faute de page!**

adresse virtuelle: 0x 7 Numéro de page
adresse physique: Numéro trame

Position dans la page
B F 0
Position dans la trame

Table des pages

| # page | # trame |
|--------|---------|
| 0 | 1 |
| 1 | 2 |
| 2 | 3 |
| 3 | 0 |
| 4 | |
| 5 | |
| 6 | |
| 7 | |

La mémoire physique est pleine, il n'y a plus de place. Que faire?

Table des pages inversée

| # trame | # page |
|---------|--------|
| 0 | 3 |
| 1 | 0 |
| 2 | 1 |
| 3 | 2 |

Fautes de page

- Une **faute de page** survient lorsque le processus a besoin d'une page qui n'est pas en mémoire.
- S'il **n'y a plus de place** dans la mémoire physique:
 1. Déterminer la page la plus anciennement utilisée (cela fait longtemps qu'on ne l'a pas utilisée);
 2. Évincer la page de la mémoire physique.
 - Cependant, cette page pourrait être utilisée à nouveau par le processus parce qu'elle peut contenir des instructions ou des données importantes.
 - Il nous faut donc sauvegarder la page sur le disque dur
 - La page est sauvegardée dans un fichier spécial, nommé le *Swap File*.

Allocation dynamique de pages

2. Effectuez la séquence d'opérations suivantes en mettant à jour les tables. Indiquez les adresses physiques correspondantes.

1. Charger valeur (e.g. LDR) à l'adresse 0x0A38
2. Écrire une valeur (e.g. STR) à l'adresse 0x3B97
3. Charger la valeur (e.g. LDR) à l'adresse 0x2928
- 4. Brancher (e.g. B) à l'adresse 0x7BF0**

Table des pages

| # page | # trame |
|--------|---------|
| 0 | 1 |
| 1 | 2 |
| 2 | 3 |
| 3 | 0 |
| 4 | |
| 5 | |
| 6 | |
| 7 | |

Table des pages inversée

| # trame | # page |
|---------|--------|
| 0 | 3 |
| 1 | 0 |
| 2 | |
| 3 | 2 |

Allocation dynamique de pages

2. Effectuez la séquence d'opérations suivantes en mettant à jour les tables. Indiquez les adresses physiques correspondantes.

1. Charger valeur (e.g. LDR) à l'adresse 0x0A38
2. Écrire une valeur (e.g. STR) à l'adresse 0x3B97
3. Charger la valeur (e.g. LDR) à l'adresse 0x2928
4. Brancher (e.g. B) à l'adresse 0x7BF0

Nombre de bits pour la position dans la page:

$$\log_2(\text{taille d'une page}) = \log_2(4 \times 2^{10}) = \log_2(2^{12}) = 12 \text{ bits}$$

| adresse virtuelle: | 0x | 7 | Numéro | Position | 4 |
|--------------------|----|--------|---------|---------------|---|
| | | | de page | dans la page | |
| adresse physique: | | Numéro | B | F | 0 |
| | | | trame | Position | |
| | | | | dans la trame | 5 |

La trame la plus anciennement utilisée est la trame 2.

Donc, on évince la page 1.

Table des pages

| # page | # trame |
|--------|---------|
| 0 | 1 |
| 1 | 2 |
| 2 | 3 |
| 3 | 0 |
| 4 | |
| 5 | |
| 6 | |
| 7 | |

Table des pages inversée

| # trame | # page |
|---------|--------|
| 0 | 3 |
| 1 | 0 |
| 2 | 1 |
| 3 | 2 |

Allocation dynamique de pages

2. Effectuez la séquence d'opérations suivantes en mettant à jour les tables. Indiquez les adresses physiques correspondantes.

1. Charger valeur (e.g. LDR) à l'adresse 0x0A38
2. Écrire une valeur (e.g. STR) à l'adresse 0x3B97
3. Charger la valeur (e.g. LDR) à l'adresse 0x2928
4. Brancher (e.g. B) à l'adresse 0x7BF0

Nombre de bits pour la position dans la page:

$$\log_2(\text{taille d'une page}) = \log_2(4 \times 2^{10}) = \log_2(2^{12}) = 12 \text{ bits}$$

| adresse virtuelle: | 0x | 7 | Numéro | Position | 4 |
|--------------------|-------|---|----------|---------------|---|
| | | | de page | dans la page | |
| adresse physique: | 0x | 2 | B | F | 0 |
| Numéro | trame | | Position | dans la trame | 5 |
| | | | | | |

On charge ensuite la page 7 dans la trame libérée (soit la trame 2).

Table des pages

| # page | # trame |
|--------|---------|
| 0 | 1 |
| 1 | |
| 2 | 3 |
| 3 | 0 |
| 4 | |
| 5 | |
| 6 | |
| 7 | 2 |

Table des pages inversée

| # trame | # page |
|---------|--------|
| 0 | 3 |
| 1 | 0 |
| 2 | 7 |
| 3 | 2 |

Fautes de page: algorithme

1. S'il **n'y a plus de place** dans la mémoire physique:
 1. Déterminer la trame la plus anciennement utilisée
 2. Évincer la page de la mémoire physique.
 1. Sauvegarder la page sur le disque dur dans un fichier spécial, nommé *Swap File*.
2. Trouver une trame libre en RAM;
3. Créer la page (ou la chercher sur le disque dur si elle a déjà été créée);
4. Placer la page en mémoire physique dans la trame sélectionnée à l'étape 2;
5. Mettre à jour la table des pages et la table des pages inversée.

Récapitulation: allocation paginée

- La mémoire:
 - **virtuelle** est divisée en **pages**;
 - **physique** est divisée en **trames**;
- Les pages et les trames ont la même taille;
- La **table des pages** stocke la correspondance entre une page et une trame;
- La traduction d'adresse implique 3 étapes:
 1. Déterminer la page de l'adresse virtuelle;
 2. Trouver la trame (dans la mémoire physique) correspondant à cette page;
 3. Remplacer le numéro de page par le numéro de trame.
- La **table des pages inversée** stocke la correspondance entre une trame et une page;
- Lorsqu'une page n'est pas chargée en mémoire, il y a **faute de page**.