

# Système d'exploitation

## SYSIR3 - SYSG4

M. Bastreggi (mba)

Haute École Bruxelles Brabant — École Supérieure d'Informatique

Année académique 2020 / 2021

D'après le cours de M.Jaumain

## Section Mémoire

### Mémoire

- pagination
- segmentation et pagination intel

## Avancement

- pagination
- segmentation et pagination intel

# Pagination - Mémoire Virtuelle

Un programme peut résider partiellement en RAM

- ▶ L'ensemble de programmes qui tournent peut dépasser la taille de la RAM physique
- ▶ Un programme à lui seul peut dépasser la taille de la RAM (obésiciel)

# Pagination

- ▶ Espace d'adressage d'un programme découpé en **pages** de taille fixe.
- ▶ Pages chargées en RAM **à la demande**
- ▶ RAM découpée en **cadres** de même taille
- ▶ Image des pages en mémoire secondaire (SWAP).

- Espace d'adressage d'un programme découpé en **pages** de taille fixe.
- Pages chargées en RAM **à la demande**
- RAM découpée en **cadres** de même taille
- Image des pages en mémoire secondaire (SWAP).

une page est un ensemble d'adresses contiguës

# Table des pages

Une table en mémoire décrit les pages de de l'espace d'adressage du programme

Présence en RAM et emplacement (no cadre).

Le registre CR3 pointe la table pour chaque processus.

La traduction est assurée par un mécanisme hardware à l'exécution (MMU).

- ▶ on doit pouvoir facilement vérifier la présence de la page
- ▶ on doit pouvoir facilement transformer l'adresse

# Système d'exploitation

## └─ Mémoire

### └─ pagination

#### └─ Table des pages

#### Table des pages

Une table en mémoire décrit les pages de de l'espace d'adressage du programme

Présence en RAM et emplacement (no cadre).

Le registre CR3 pointe la table pour chaque processus.

La traduction est assurée par un mécanisme hardware à l'exécution (MMU).

- on doit pouvoir facilement vérifier la présence de la page
- on doit pouvoir facilement transformer l'adresse

D'autres tables du noyau sont utilisées pour retrouver les pages sur le disque



# Défaut de page (page fault)

Une référence à une page absente provoque un basculement dans l'OS.

Exception "défaut de page" (page fault).

Cela permet le chargement de la page.

Il sera nécessaire de réexécuter l'instruction qui a provoqué le défaut de page.

# Système d'exploitation

- └─ Mémoire

- └─ pagination

- └─ Défaut de page (page fault)

## Défaut de page (page fault)

Une référence à une page absente provoque un basculement dans l'OS.

Exception "défaut de page" (page fault).

Cela permet le chargement de la page.

Il sera nécessaire de réexécuter l'instruction qui a provoqué le défaut de page.

l'adresse de l'instruction est mémorisée dans CR2 sur intel

# Taille des pages et traduction d'adresse

Une page d'un espace d'adressage est chargée dans un cadre mémoire.

- ▶ pages - numérotées depuis 0 au sein de l'espace d'adressage
- ▶ cadres - numérotés depuis 0 au sein de la RAM.

# Taille des pages et traduction d'adresse

**Numéro de page** pour une adresse  $ad$  :  $ad \text{ DIV Taille des pages}$

**Offset dans la page** :  $ad \text{ MOD Taille des pages}$

**Adresse d'un cadre** :  $n^{\circ}\text{cadre} * \text{Taille d'une page}$

$\text{adresse physique} = \text{adresse cadre} + \text{offset}$

# Taille des pages et traduction d'adresse

Si la taille des pages est une puissance de deux, cela facilite la traduction des adresses virtuelles.

Adresse virtuelle – page de taille  $2^{\text{exp } n}$

n° de page dans l'espace d'adressage virtuel	offset dans la page n bits de droite
<b>DIV (<math>2^{\text{exp } n}</math>)</b>	<b>MOD (<math>2^{\text{exp } n}</math>)</b>

- ▶ partie gauche de l'adresse = n° de page
- ▶ partie droite de l'adresse = offset dans la page

# Taille des pages

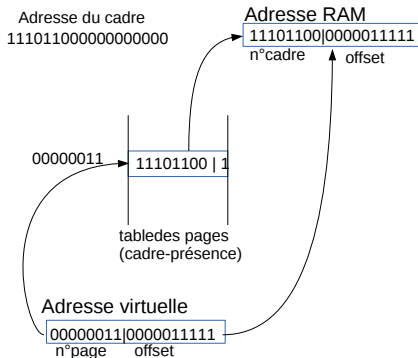
Si le choix de la taille de page est de la forme  $2^n$  ?



les  $n$  bits de droite de l'adresse indiquent l'offset,  
à gauche le numéro de cadre

# Traduction d'adresse et taille des pages

Traduction d'adresse avec pages de taille  $2^{\text{exp } 10}$



# Questions ?





# exemple réduit - Espace d'adressage des programmes

Adresse virtuelle sur 16 bits, pages de 4Kib :

adresse virtuelle : 4 bits pour le n° de page, 12 bits pour le décalage

- ▶ espace d'adressage et mémoire divisés en pages/cadres de 4Kib ( $2^{12}$ ).
- ▶ 4 bits pour le numéro de page (16 valeurs 0-15)
- ▶ espace adressage de 64Kib ( $16 \times 4\text{Kib}$ ).

dans ce cas un programme pourrait adresser jusqu'à 64Kib

# exemple réduit - RAM

soit une mémoire physique de 32Kib

- ▶ L'espace d'adressage est plus étendu que la mémoire (obésiciels).
- ▶ 32Kib en mémoire physique (page de 4Kib => 8 cadres).
- ▶ La correspondance entre les pages du programme et les cadre en RAM est assurée par la **Table des Pages**.
- ▶ Un registre contient l'adresse de la table

# exemple réduit : table de pages

correspondance entre adressage virtuel et physique

Table des pages

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

N° de page virtuel

N° de cadre de page

bit présence

cadres en mémoire

7	page n°11
6	page n°2
5	page n°9
4	page n°4
3	page n°5
2	page n°0
1	page n°1
0	page n°3

N° de cadre en mémoire

# exemple réduit suite

- ▶ Les pages absentes sont chargées en mémoire (cadre) sur demande.
- ▶ Une copie des pages du programme est sur disque.
- ▶ Les transferts entre disque et mémoire se font par page/cadre.
- ▶ On veillera à sauvegarder sur disque les pages modifiées.

# traduction d'adresse

MOV reg, [8192]	devient	MOV reg, [24576]
MOV reg, [20500]	devient	MOV reg, [12308]
MOV reg, [32780]	devient	?

adresse	adresse en binaire (page décalage)	page	décalage	cadre	adresse physique
8192	10 000000000000	2	0	6	110 000000000000 (=24576)
20500	101 000000010100	5	20	3	11 000000010100 (=12308)
32780	1000 000000001100	8	12	absente	?

# défaut de page

Un défaut de page (page 8) provoque une exception

Rôle de l'OS en cas de défaut de page ?

- ▶ choisir une **victime** (page peu utilisée à supprimer de la RAM).
- ▶ **sauvegarder** la page victime sur le disque si modifiée.
- ▶ **charger** la page référencée, à la place de la victime.
- ▶ mettre à jour la **table** des pages.
- ▶ RIP = adresse instruction à réexécuter

# Système d'exploitation

## └─ Mémoire

### └─ pagination

#### └─ défaut de page

Un défaut de page (page 8) provoque une exception

Rôle de l'OS en cas de défaut de page ?

- choisir une **victime** (page peu utilisée à supprimer de la RAM).
- **sauvegarder** la page victime sur le disque si modifiée.
- **charger** la page référencée, à la place de la victime.
- mettre à jour la **table** des pages.
- RIP = adresse instruction à réexécuter

sur intel RIP = CR2 pour réexécuter

# défaut de page

On choisit la page 9 comme victime, on peut réutiliser le cadre 5

9	5	1
8	0	0

la page 8 est maintenant présente dans le cadre 5 :

9	0	0
8	5	1



# exemple réduit suite

- ▶ pages de 4Kib - 12 bits
- ▶ adresse linéaire 64Kib (16 bits)
- ▶ 16 bits (4 et 12 bits) (16 pages)
- ▶ adresse physique (déposée sur bus d'adresse) 32Kib
- ▶ 15 bits (3 et 12 bits) (8 cadres)

# défaut de page

- ▶ No de page virtuel (4 bits) est l'index dans la TP
- ▶ Si page absente déroutement (interruption) vers l'OS
- ▶ Sinon remplacer les 4 bits de poids fort par les trois du numéro de cadre

# exemple réduit suite

- ▶ écrire la représentation binaire des adresses suivantes et les traduire conformément à la TP fournie.
- ▶ 8192 (2,0) (2000h) devient 24576 (6,0) (6000h)
- ▶ 20500 (5,20) (5014h) devient 12308 (3,20) (3014h)
- ▶ 32780 (8,12) (800Ch) devient 20492 (5,12) (500Ch)

## Système d'exploitation

└─ Mémoire

└─ pagination

└─ exemple réduit suite

- écrire la représentation binaire des adresses suivantes et les traduire conformément à la TP fournie.
- 8192 (2,0) (2000h) devient 24576 (6,0) (6000h)
- 20500 (5,20) (5014h) devient 12308 (3,20) (3014h)
- 32780 (8,12) (800Ch) devient 20492 (5,12) (500Ch)

Décimal	Binaire	Hexadécimal
8192	10 0000 0000 0000	2 000
24576	110 0000 0000 0000	6 000
20500	101 0000 0001 0100	5 014
12308	11 0000 0001 0100	3 014
32780	1000 0000 0000 1100	8 00C
20492	101 0000 0000 1100	5 00C

# Intel 32 bits

Sur intel, la traduction des adresses est confiée au circuit M.M.U.

C'est également lui qui génère l'exception "défaut de page"

Le registre **CR3** contient l'adresse de la table de pages

# Intel 32 bits

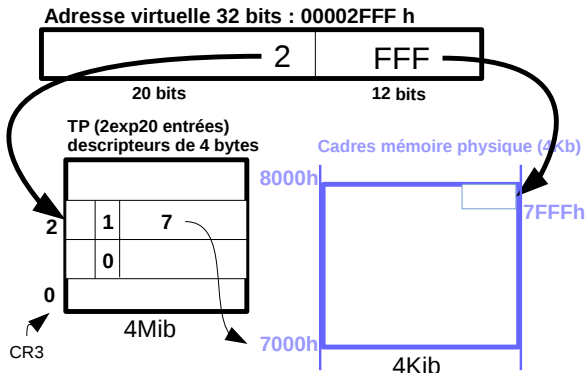
adresse sur 32 bits et pages de 4Kib

mémoire virtuelle, structure d'une adresse

n° de page dans l'espace d'adressage virtuel	décalage dans la page
20 bits	12 bits

# table des pages : exemple

mémoire virtuelle, pagination exemple



2FFF est traduit en 7FFF

# Questions

- ▶ A quel moment a lieu la traduction d'adresse ?
- ▶ Si une instruction prend 1 ns, la consultation de la TP peut-elle prendre autant ?
- ▶ Quelle est la limite de la taille d'un process sur un intel 32 bits ?



# taille de la table ?

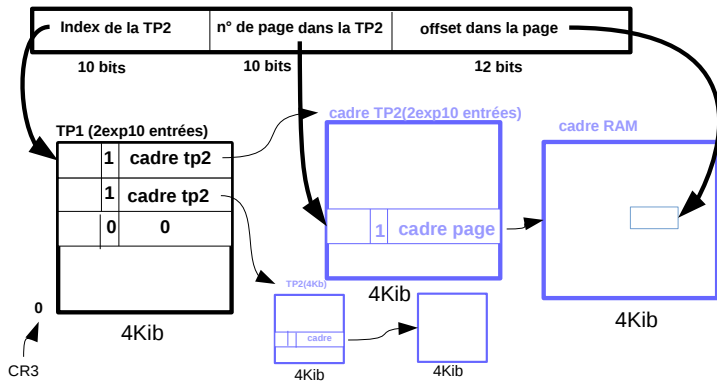
Table des pages en RAM : taille ?

- ▶ Adresse sur 32 bits (pour adresser 4Gib)
- ▶ Pages de 4Kib (12 bits de décalage et 20 de numéro de page)
- ▶ taille de la table  $2^{20}$  entrées de 4 bytes (4Mib et une par process !)

# table des pages à deux niveaux (paginée)

mémoire virtuelle, pagination à deux niveaux

adresse sur 32 bits



# tables des pages : tailles

- ▶ Une table des pages de niveau 1 par process (1024 entrées) (4Kib)
- ▶ 1024 Tables de pages de niveau 2 (4Mib au total)
- ▶ Ces dernières ne résident pas en mémoire de manière permanente

# Système d'exploitation

## └─ Mémoire

### └─ pagination

#### └─ tables des pages : tailles

- Une table des pages de niveau 1 par process (1024 entrées) (4Kib)
- 1024 Tables de pages de niveau 2 (4Mib au total)
- Ces dernières ne résident pas en mémoire de manière permanente

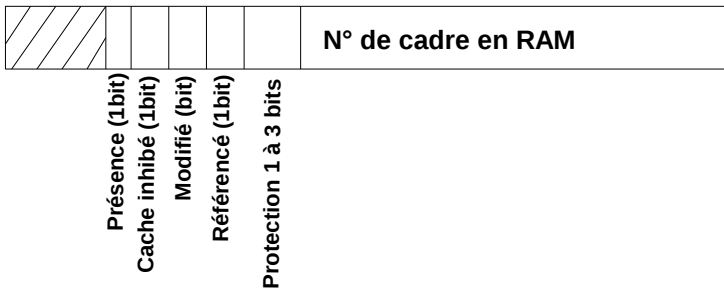
Les intel 64 bits permettent d'adresser 256 Tib, avec une hiérarchie de tables de pages à trois niveaux

adresses virtuelles : 9bits - 9 bits - 9 bits - 9 bits - 12 bits

# table des pages : entrée

un exemple du contenu de l'entrée d'une table de pages

exemple d'entrée de la table de pages



# Questions ?



# Remplacement de page

- ▶ Au moment de tradure une adresse, le MMU décèle une absence de page et provoque l'interruption PAge Fault.
- ▶ L'OS gère l'interruption et en cas de mémoire pleine va devoir faire de la place pour la nouvelle page en choisissant une page à évincer de la mémoire.
- ▶ Comment choisir la page à évincer ?
- ▶ La meilleure victime est celle qui provoquera un nouveau défaut de page le plus tard ...

# Hasard.

Retirer une page au hasard. Cette technique est très rapide mais la page retirée pourrait être redemandée très rapidement.

Le nombre de défauts de page pourrait être trop grand. Il existe probablement une meilleure technique.



# La page idéale.

La meilleure page à éliminer sera celle qui sera redemandée après le plus grand nombre d'instructions exécutées.

Il est impossible d'identifier cette page.

Mais on peut se dire qu'une page peu utilisée ou qui n'a plus été utilisée depuis longtemps ne sera probablement pas utilisée rapidement.

Deviner le futur en regardant le passé !

# bits R et M

- ▶ Utilité des bits R (récemment référencée) et M (modifiée)
- ▶ R : référencée **depuis la dernière interruption horloge**
- ▶ M : modifiée **depuis son chargement** (à réécrire sur le disque)
- ▶ Le MMU met ces bits à 1 (lire  $R \leftarrow 1$ , écrire  $M \& R \leftarrow 1$ )
- ▶ L'OS modifie M et R (chargement de la page ( $M \leftarrow 0$ ,  $R \leftarrow 1$ ), interruption horloge ( $R \leftarrow 0$ ))

# NRU - Not Recently Used

Choisit une page qui n'a pas été récemment référencée

# NRU - bits R et M

## Not Recently Used

classe	R	M	
0	0	0	non récemment référencée, non modifiée
1	0	1	non récemment référencée, modifiée
2	1	0	récemment référencée, non modifiée
3	1	1	récemment référencée, modifiée

# NRU - Not Recently Used

- ▶ Choisit au hasard une page dans la classe de plus petite valeur.
- ▶ Algorithme simple.
- ▶ Performances acceptables.

NB. une page modifiée (il y a longtemps) sera choisie plutôt qu'une récemment référencée

# Système d'exploitation

└─ Mémoire

└─ pagination

└─ NRU - Not Recently Used

- Choisit au hasard une page dans la classe de plus petite valeur.
- Algorithme simple.
- Performances acceptables.

NB. une page modifiée (il y a longtemps) sera choisie plutôt qu'une récemment référencée

part du principe qu'une page modifiée mais non récemment référencée à moins de chances d'être nouvellement référencée qu'une page récemment référencée

# NRU

- ▶ Quels sont les avantages et les défauts de cette technique ?
- ▶ Quand et pourquoi R et M sont-ils modifiés ?
- ▶ Pourquoi remettre R à 0 régulièrement ?
- ▶ Pourquoi ne pas remettre M à zéro au même moment ?

# FIFO - First In First Out

- ▶ A tour de rôle.
- ▶ Les plus anciennes pages sont évincées
- ▶ Algorithme simple.
- ▶ Mémoriser l'ordre de chargement des pages
- ▶ Performances médiocres.
- ▶ Finit par évincer des pages importantes.



# FIFO - Seconde Chance Out

Tient compte de l'utilisation récente de la page

- ▶ Si la première a son bit  $R=1$  alors  $R=0$  et réinjecte en queue
- ▶ Tous les bits à 1  $\rightarrow$  FIFO pur
- ▶ Performances acceptables.

# LRU - Least Recently Used

- ▶ La victime est la page non utilisée depuis le plus longtemps.
- ▶ Algorithme coûteux en temps.

Maintenir une liste triée des pages. Mis à jour à chaque référence de page

Il existe des solutions plus matérielles à cet algorithme.

# LRU - compteur

Utilise un **registre spécial** compteur d'instructions

- ▶ Le compteur d'instructions (64 bits) est inscrit dans la TP et mis à jour à chaque accès

LRU est celle avec le plus petit compteur  
(-) il faut ce registre hardware

# Système d'exploitation

## └─ Mémoire

### └─ pagination

### └─ LRU - compteur

Utilise un **registre spécial** compteur d'instructions

- Le compteur d'instructions (64 bits) est inscrit dans la TP et mis à jour à chaque accès

LRU est celle avec le plus petit compteur

(-) il faut ce registre hardware

## Un compteur software dans la table des process par exemple

# NFU - Not Frequently Used

Simulation software de LRU

- ▶ à chaque interruption horloge on ajoute R à un compteur initialisé à 0

Victime par NFU est celle avec le compteur plus petit

(-) n'oublie pas le passé (page très utilisée au début et après plus ...)

# AGING - amélioration de NFU

Addition en deux temps :

- ▶ shift de 1 à droite
- ▶ bit R ajouté à gauche

Victime par AGING est celle avec le compteur plus petit

Avec un compteur de  $n$  bits permet de garder l'historique des  $n$  dernières interruptions horloge

# Exercice.

Un ordinateur possède une mémoire de 32K, des adresses virtuelles sur 16 bits et des pages de 8K. (a) Construisez un système de pages à un niveau, complétez-le avec des valeurs cohérentes et donnez les adresses physiques correspondantes aux adresses virtuelles 0080h, 0800h et 8000h. (b) Même question avec un système à deux niveaux et des pages de 4K.

# Exercice

Un ordinateur possède 4 cadres mémoire. On donne les moments de chargement, les moments de dernier accès, les bits R et M des 4 pages présentes

page	chargement	accès	R	M
0	126	279	1	0
1	230	260	0	0
2	128	272	0	0
3	160	280	1	1

Quelle sera la page remplacée par NRU ?, FIFO ?, LRU ?



## Système d'exploitation

## └─ Mémoire

## └─ pagination

## └─ Exercice

## Exercice

Un ordinateur possède 4 cadres mémoire. On donne les moments de chargement, les moments de dernier accès, les bits R et M des 4 pages présentes

page	chargement	accès	R	M
0	126	279	1	0
1	230	260	0	0
2	128	272	0	0
3	160	280	1	1

Quelle sera la page remplacée par NRU ?, FIFO ?, LRU ?

NRU 1 ou 2 FIFO 0 LRU théorique 1

# Exercice

Si on utilise FIFO avec 4 cadres et 8 pages, combien de défauts de page se produira-t-il si la liste des références aux pages est 0172327103 et si tous les cadres sont initialement vides ? Et avec LRU ?

Si on utilise FIFO avec 4 cadres et 8 pages, combien de défauts de page se produira-t-il si la liste des références aux pages est 0172327103 et si tous les cadres sont initialement vides ? Et avec LRU ?

6 et 7

# Questions ?



# Working Set

Un nouveau processus est élu : il provoque quelques défauts de pages jusqu'à ce que son 'working set' soit en mémoire. Un système à temps partagé provoque des échanges de processus et éventuellement, une copie du working set sur le disque. Si le processeur passe son temps à recharger le working set, il y a écrasement du système. Certains S.E. n'attendent pas les défauts de page en cascade mais rechargent le working set d'un processus.

# Locale ou globale

Changer de page : la LRU du processus ? de la mémoire ?  
Il est plus performant de considérer toute la mémoire.  
Mais combien de page faut-il donner à un processus ?  
Est-ce fixe ? Surveiller le nombre de défaut de page des processus (trop élevé, il ne dispose pas d'assez de mémoire, trop petit, il dispose de trop de mémoire)

# Taille des pages

Trop grande ? perte de place

Trop petite ? grande table et temps de chargement long

En pratique, optimisé pour le transfert avec le disque = 4K.

# E/S et pagination

E/S avec DMA, le processus 1 lance le transfert DMA dans la page X et est bloqué.

Processus 2 prend la main et demande une page : si la page X est sélectionnée pour être retirée et son adresse donnée au processus 2, cette page se remplira du transfert DMA !

(Solutions : soit verrouiller ces pages, soit laisser au S.E. les buffers d'E/S)



# Exercice

Refaire la chronologie d'un défaut de page.

4000 MOV AX,[6000]

4003 ...

# Localité et performance

- ▶ les pages utilisées lors des  $K$  dernières références a tendance à se stabiliser (working set)
- ▶ le MMU dispose d'une mémoire associative ( $n^{\circ}$  page, M, protection, cadre) TLB pour 256 pages

## Avancement

- pagination
- segmentation et pagination intel

# segmentation et pagination intel

## Notion d'espace d'adressage

- ▶ segmentation mode protégé
- ▶ pagination

# mode protégé

En présence de pagination

- ▶ La traduction d'adresse se fait en plusieurs étapes
- ▶ adresse logique -> adresse linéaire -> adresse physique

# rappel : mode protégé (386)

on part d'une **adresse logique**

- ▶ Une **adresse logique** est composée de deux parties : **sélecteur-offset**
- ▶ **registres sélecteurs** : CS, DS, SS, ...
  - CS - associé au segment de code
  - DS - associé au segment de données
  - SS - associé au segment de pile
  - ...

# rappel : adresse logique

Un programme utilise plusieurs segments, leur utilisation est souvent implicite :

- ▶ JMP BOUCLE - JMP **CS** :BOUCLE
- ▶ MOV EAX,[EBX] - MOV EAX, [**DS** :EBX]
- ▶ PUSH EAX - utilise **SS** :ESP

# rappel : adresse logique

- ▶ **sélecteur** de segment : 16 bits (CS, DS, SS,...)
- ▶ **offset** dans le segment : 32 bits (offset) .

Chaque segment est un espace d'adressage délimité par une base et une limite (taille)

Taille maximum d'un segment : 4Gib



# descripteur de segment

Sans pagination **l'adresse calculée** est une adresse physique en RAM

# Système d'exploitation

## └─ Mémoire

### └─ segmentation et pagination intel

#### └─ descripteur de segment

Sans pagination l'**adresse calculée** est une adresse physique en RAM

En présence de pagination, c'est une adresse dans l'espace d'adressage du programme

# segmentation et pagination sur Intel

- ▶ Intel combine segmentation et pagination.
- ▶ L'adresse de base du segment est une adresse linéaire dans un espace unique paginé.
- ▶ Une table des pages (deux niveaux) est associée à cet espace.
- ▶ Un segment peut résider partiellement en RAM

# segmentation et pagination

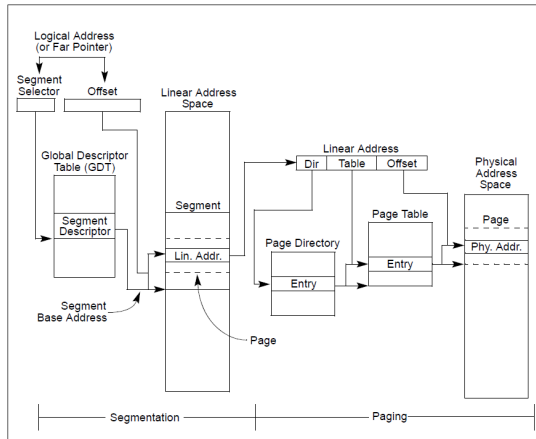


Figure 3-1. Segmentation and Paging





# Questions ?



shutterstock - 104523281

# images

toutes les images du chapitre segmentation sont issues de  
intel 64 and I1-32 Architectures Software Developer's Manual (2011)

-  Modern Operating Systems Fourth edition - Andrew Tanenbaum, Herbert Bos - Pearson Education
-  Advanced Programming in the UNIX Environment Third Edition - W.Richard Stevens, Stephen A. Rago - Addison Wesley (2014)
-  Programmation Système en C sous Linux 2ième édition - Christophe Blaess - Eyrolles (2005)
-  Intel 64 and IA-32 Architectures Software Developer's Manual - December (2011) (pour toutes les images du chapitre mémoire)

# remerciements

merci à P.Bettens et M.Codutti pour la mise en page



# Crédits

Ces slides sont le support pour la présentation orale des activités d'apprentissage **SYSIR3** et **SYSG4** à HE2B-ÉSI

## Crédits

La distribution opensuse  
du système d'exploitation **GNU Linux**.

**LaTeX/Beamer** comme système d'édition.

**GNU make, rubber, pdfnup**, ... pour les petites tâches.

## Images et icônes

deviantart, flickr, The Noun Project 