Gestion de la Mémoire

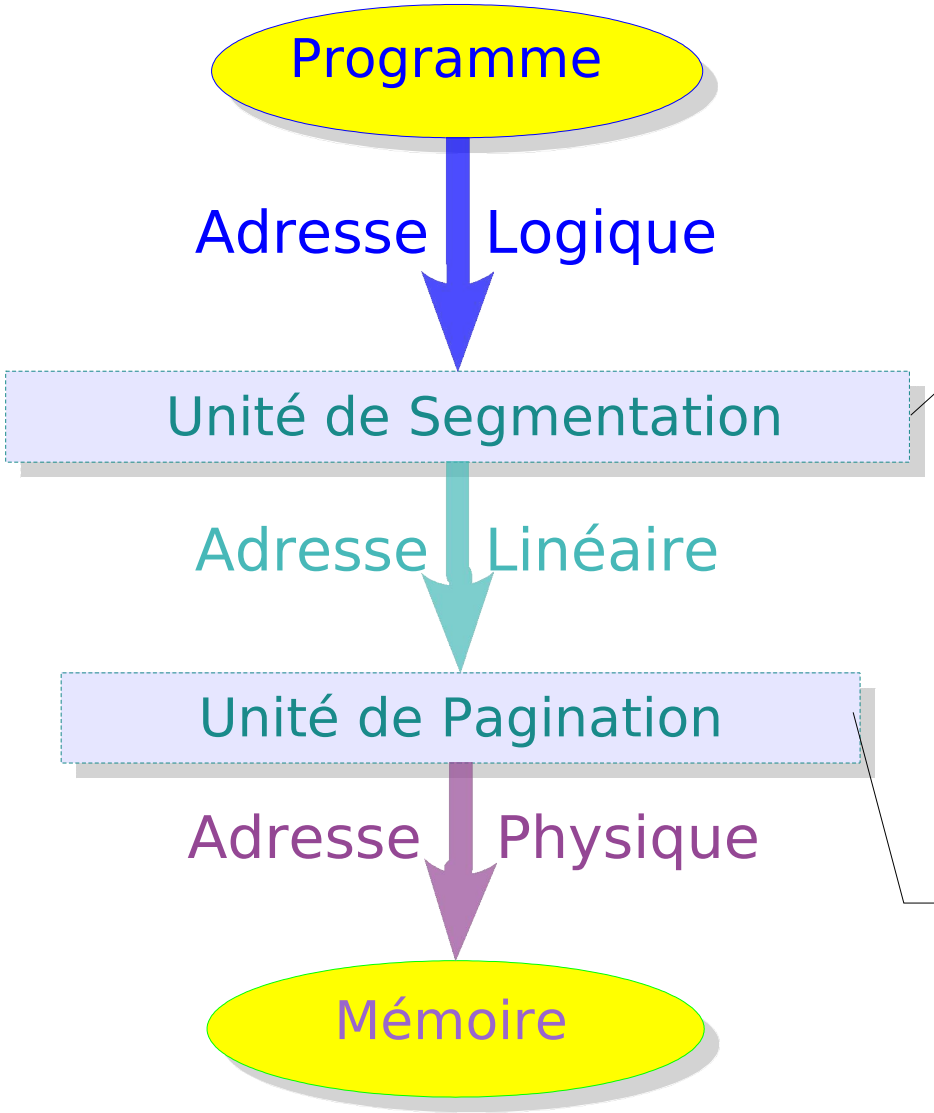
Adressage Mémoire

- Le modèle d'adressage intel x86
- Segmentation
  - Au niveau matériel
    - Registres
    - Descripteurs
  - Mise en oeuvre Linux
- Gestion des Caches
  - Caches matériel
  - TLB
- Pagination
  - Au niveau matériel
    - Pagination normale
    - Pagination étendue
    - Pagination à 3 niveaux
    - Adressage physique étendu (PAE)
  - Mise en oeuvre par Linux
    - Tables de Pages
    - Cadres réservés
    - Tables noyau et processus

Adressage intel 80x86

- 3 niveaux d'adresses
  - Adresses Logiques (« segmentées »)
    - segment + offset
    - Les adresses manipulées par les programmes en langage machine
  - Adresses Linéaires (ou virtuelles)
    - Entier 32 bits : jusqu'à 4 Md d'@ (0x00000000 à 0xffffffff)
  - Adresses Physiques
    - Entier 32 bits aussi : le signal éléctrique qui est placé sur le bus

Translation d'adresses



2 modes :
- réel
- protégé (>=386)

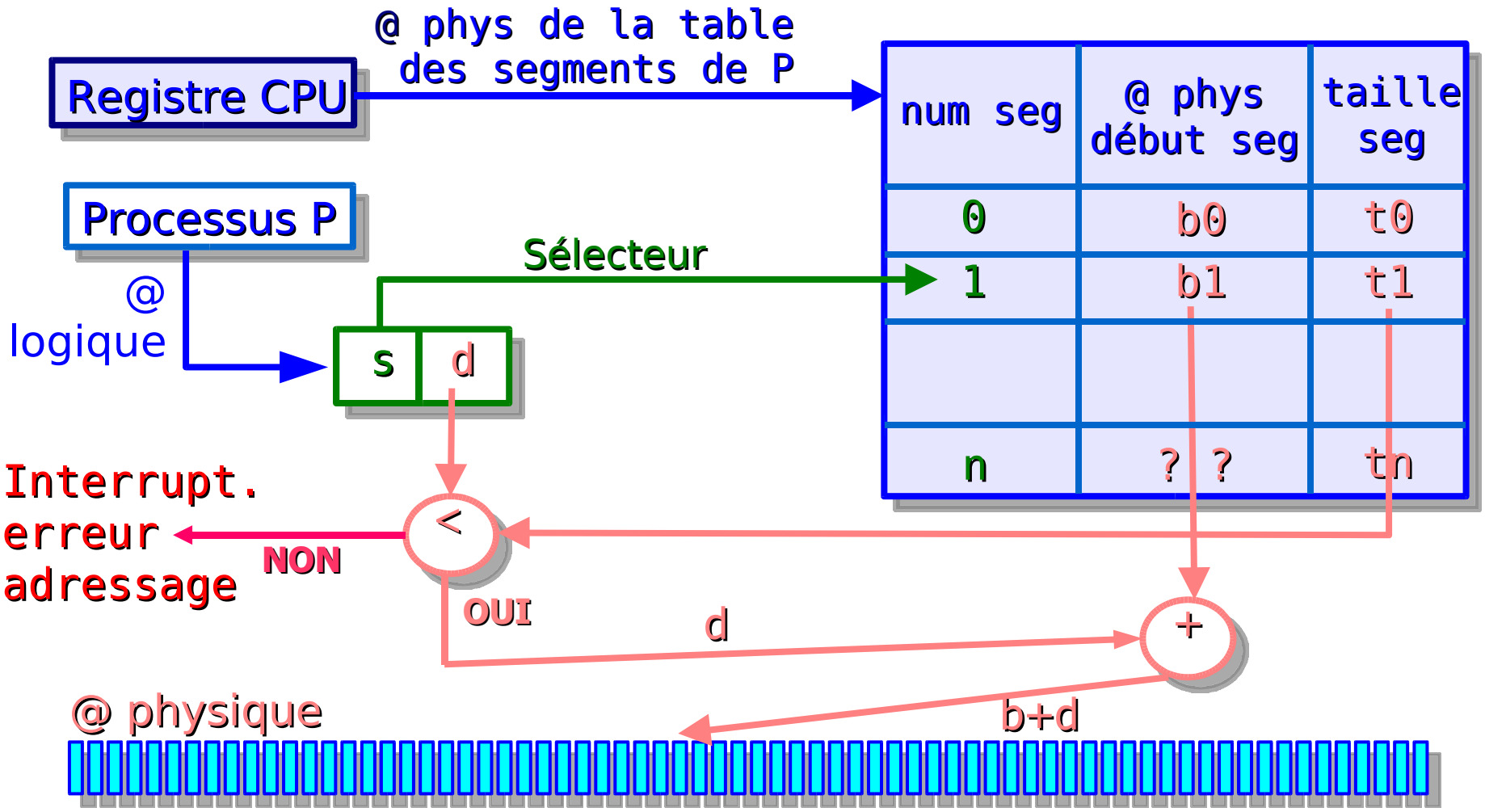
Basique (regular) :
- pagination normale à 2 niveaux

Evolutions :
- 586 : pagination étendue (1 niv.)
- PPro : Extension. d'@ Phys. (PAE)
- PIII : Extension. Page (PSE-36)

Principe de la segmentation

- Hypothèses : le compilateur a généré un exécutable constitué de segments
  - Taille d'un segment << taille MC
- Un segment = un numéro + une taille
- Une adresse logique = (s,d)
  - s = numéro du segment
  - d = déplacement dans le segment (offset)
- Chaque processus a une table des segments
  - Doit être en MC lorsque le processus est **prêt**
  - Sinon, elle peut être stockée en zone de swap (sur le disque)

Schéma (basique) de fonctionnement



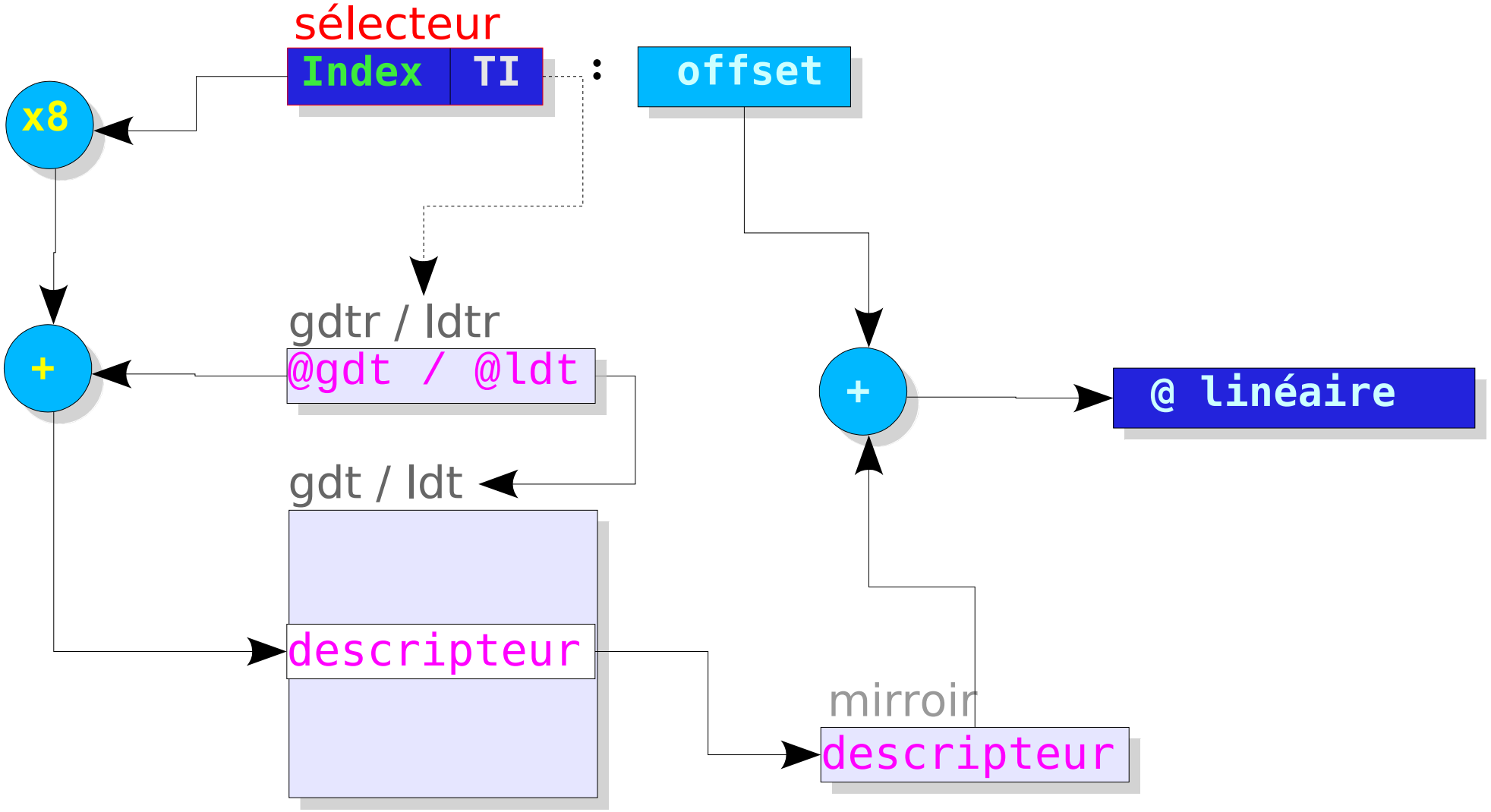
Registres de segmentation sur 80x86

- 3 registres de sélection spécialisés :
  - CS : désigne le segment de code
    - Autre fonction : fixe le niveau de privilège du CPU
  - SS : désigne le segment de pile (stack)
  - DS : désigne le segment de données
- 3 registres de sélection généraux :
  - ES, FS, GS
- Mais aussi des registres de support :
  - GDTR, LDTR (tables de descripteur)
  - 6 registres « miroir » des descripteurs (non prog.)

Mise en oeuvre de la segmentation sur 80x86

- @logique = [selecteur de segment]:[offset]
  - Sélecteur = 16 bits (dans reg CS, DS, SS, ES, FS, GS)
    - 13 bits index (0 - 8191)
    - 1 bit TI : Table Indicator
    - (2 bits RPL : niveau de privilège)
  - Offset = 32 bits (maximum 4 Go)
- Index : numéro du Descripteur de Segment dans la Table des Descripteurs
  - Descripteur de Segment : Structure de 8 octets (...)
  - Table des Descripteurs : 2 possibilités selon TI :
    - Table Globale (GDT), désignée par reg. gdtr
    - Table Locale (LDT), désignée par reg. ldtr

Schéma de fonctionnement sur 80x86



Descripteurs de segments sur 80x86

- Base = 32 bits : @linéaire du 1er octet du segment
- G = 1 bit : granularité (taille seg multiple de 1 ou 4096)
- Limit = 20 bits : longueur seg (Max 1 Mo/4 Go selon G)
- S = 1 bit : Système flag (données système ou « normales »)
- Type = 4 bits
  - CSD : Code Seg. Desc. (GDT ou LDT)
  - DSD : Data Seg. Desc. (GDT ou LDT)
  - TSSD : Task State Seg. Desc. (sauvegarde contexte) (GDT)
  - LDTD : Local Desc. Table Desc. (seg. contenant LDT) (GDT)
- DPL = 2 bits : Desc. Privilège Level (niveau CPL requis)
- Segment-present = 1 bit (swapin/swapout)
- D/B flag = 1bit : offset sur 16 ou 32 bits

Segmentation : mise en oeuvre sous Linux

- Segmentation et Pagination : des mécanismes redondants !
  - Contrainte Unix (Linux) : portabilité
    - Support Seg. limité sur certaines architectures (RISC)
  - Gestion plus simple avec seulement la Pagination
- Stratégie Linux : support minimal de la Segmentation
  - Cas général : même espace d'adressage logique pour tous les processus
    - Nombre total de segments limité
    - Sauvgarde des desc. de segments uniquement dans GDT
  - Cas particuliers (ex: WINE) : A. S. **`modifiy_ldt()`**

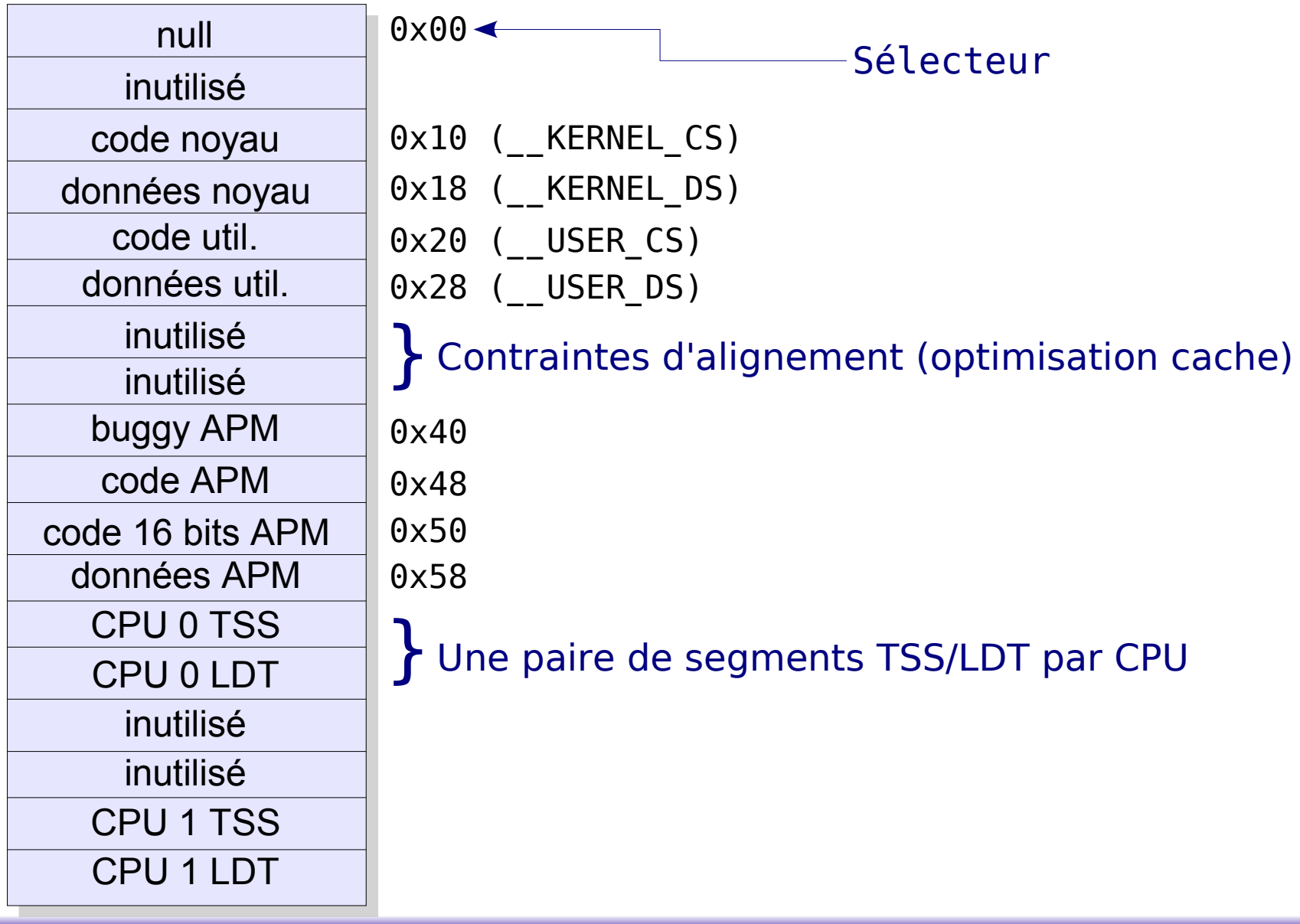
Les segments utilisés par Linux

- Segments du Noyau
  - Code :
    - base= 0x00000000, limit=0xfffff, G=1 (limite exprimée en pages) : adresses de 0 a 2^32-1
    - DPL=0 : CPL requis = 0 (mode noyau)
    - Type = 0xA : permission RX
  - Data :
    - idem Code mais type=2 (perm=RW) au lieu de 0xA
- Segments utilisateurs
  - Code et Data :
    - idem Kernel mais DPL = 3

Les segments utilisés par Linux (2)

- 1 segment TSS (Task State Seg.) par CPU
  - Stockés sequentiellement dans le tableau **`init_tss`**
  - Base : nieme CPU = nieme entrée de **`init_tss`**
  - G=0, Limit = **`0xeb`** (236 octets)
- 1 segment LDT par défaut (**`default_ldt`**)
  - LDT commune par défaut à tous les processus
  - Contient un unique descripteur : le desc. Nul
  - Possibilité de remplacer ce segment LDT par défaut par un segment construit par le processus
- 4 Segments pour BIOS APM (2 code, 2 data)

Table Globale des Desc. de Segments de Linux (GDT)

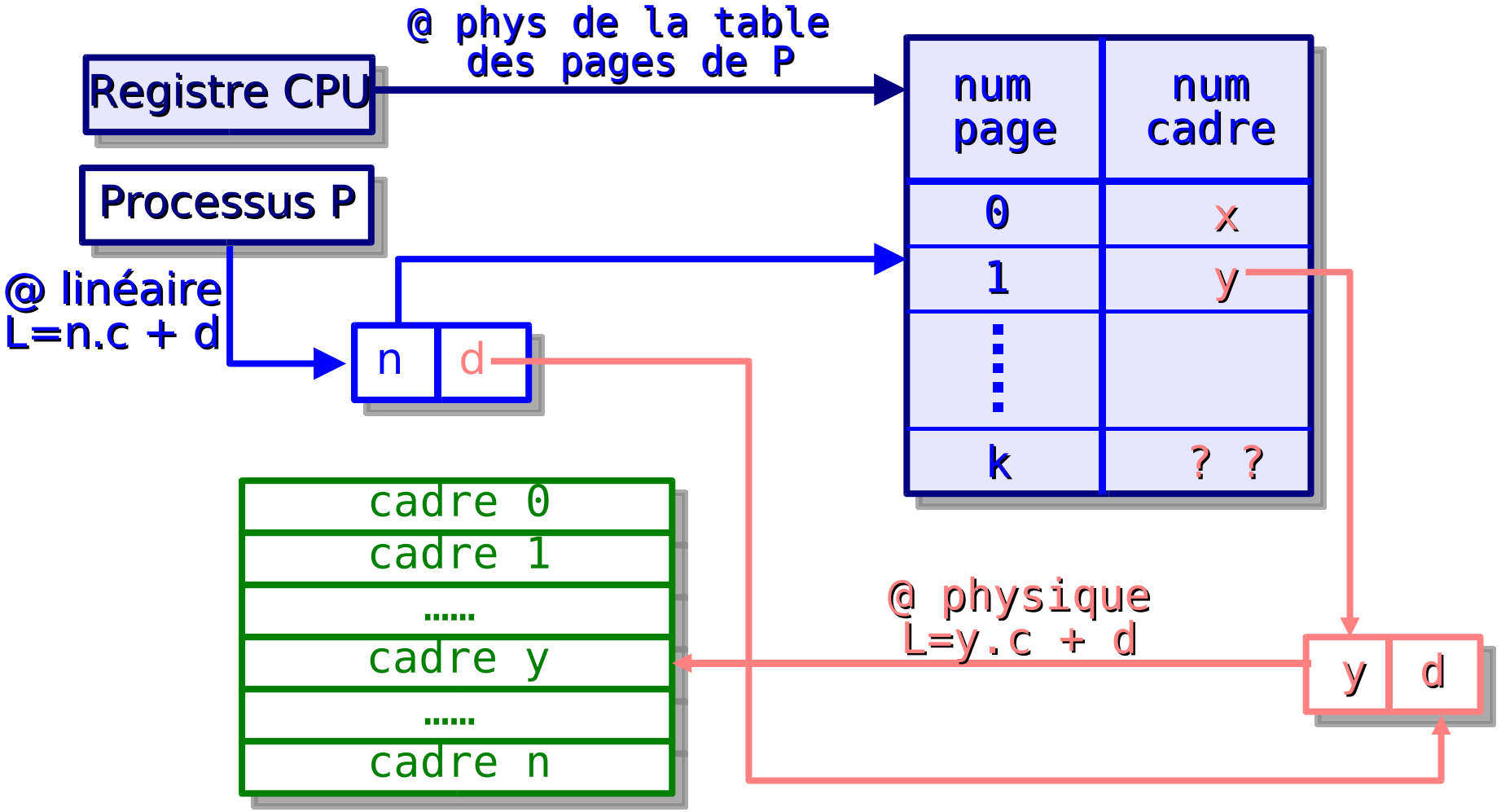


Principes généraux de la pagination

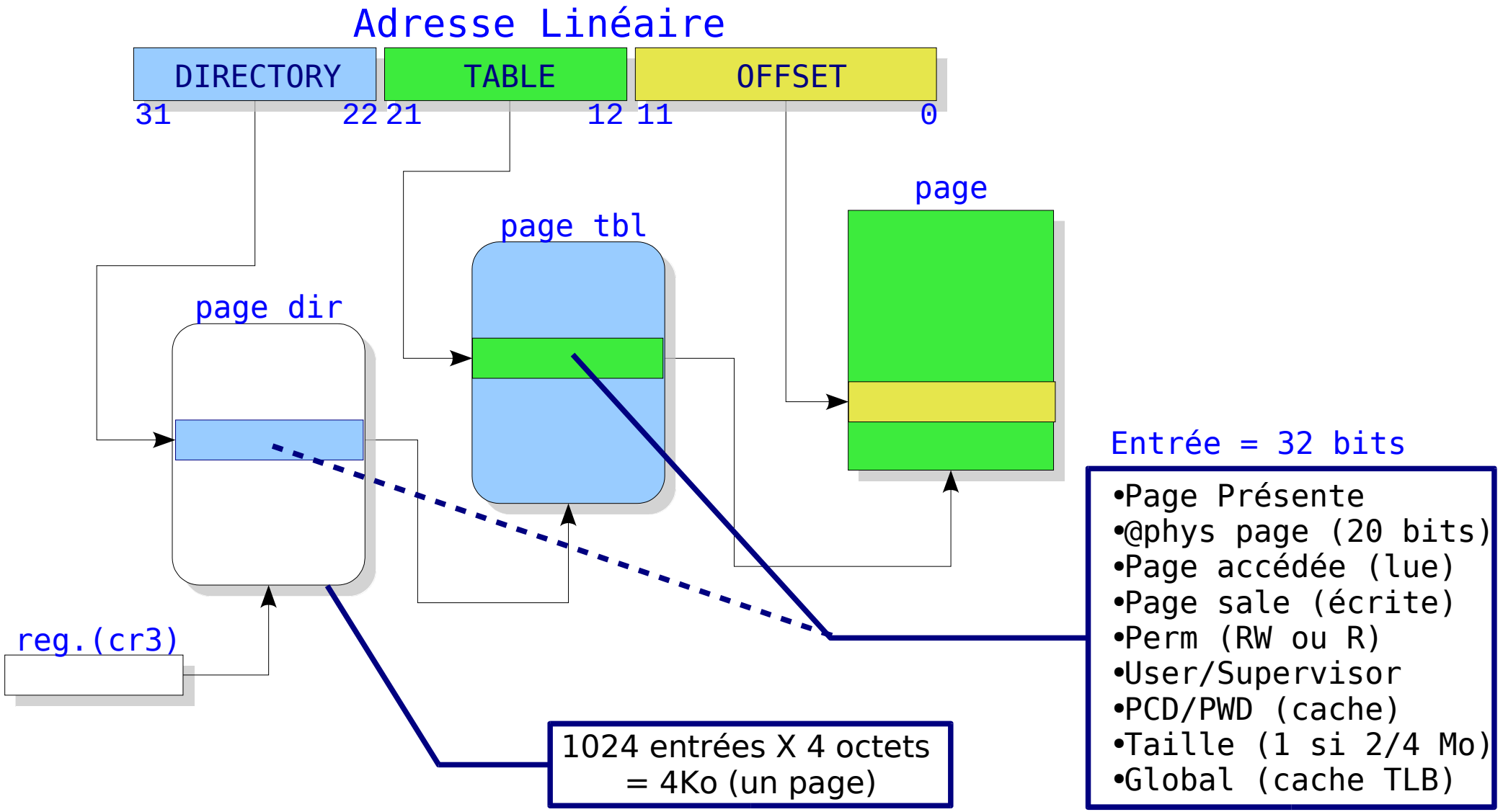
- La mémoire physique est découpée en **cadres** (frames)
  - Taille constante **c** (linux/x86 : **c = 4 Ko**)
- Espace logique est découpé en **pages**
  - Taille constante **c** aussi :
    - les pages sont placées dans des cadres
- Le compilateur n'a pas de rôle spécial
  - Il engendre un espace d'adressage logique linéaire
  - La conversion d'une adresse logique L sous la forme L=(n,d) est **implicite**
    - c = taille de page
    - **n** = L div c : numéro de la page
    - **d** = L mod c : déplacement dans la page

    ⇒ L = **n**.c + **d**

Schéma (basique) de Fonctionnement



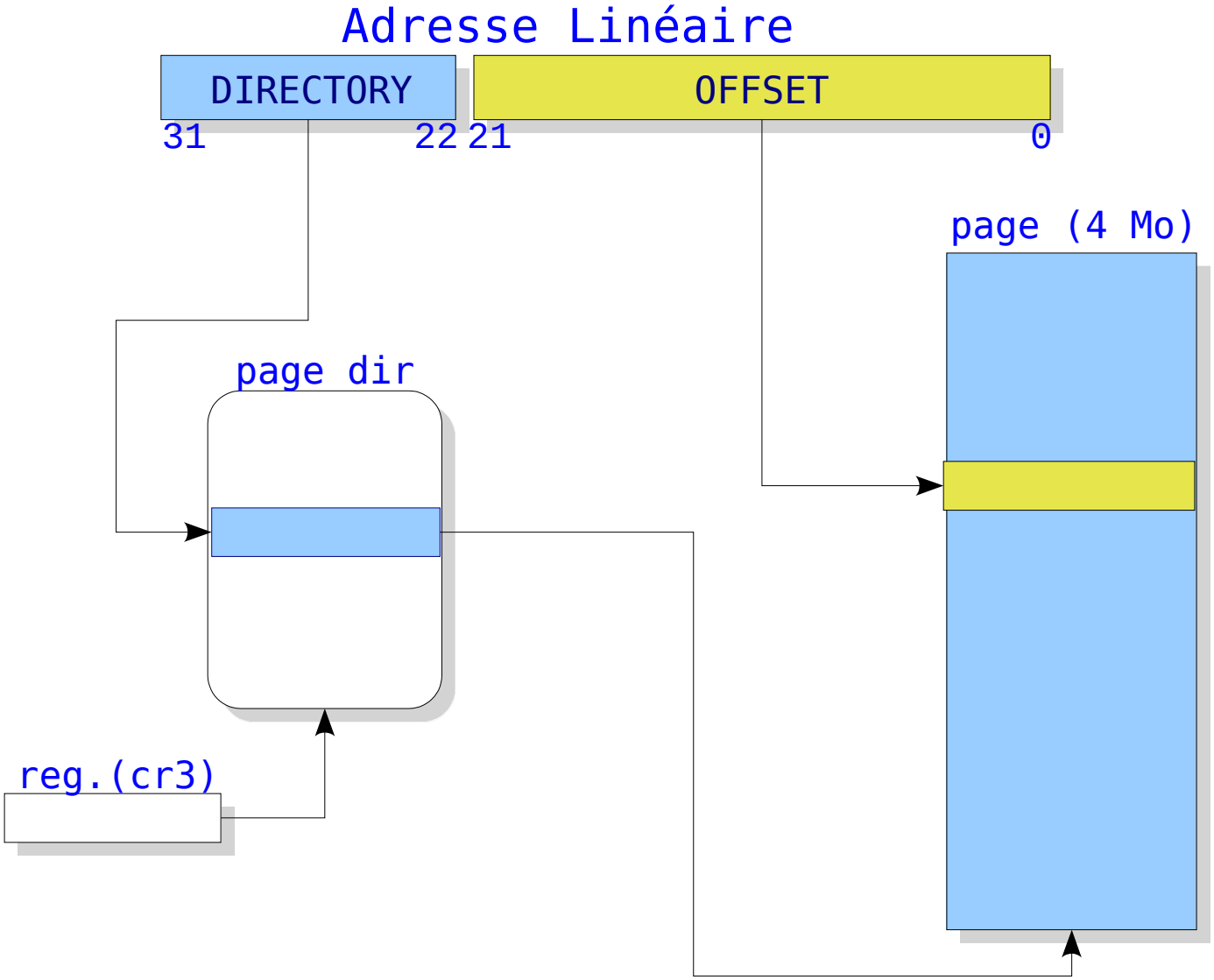
La pagination « normale » des 80x86



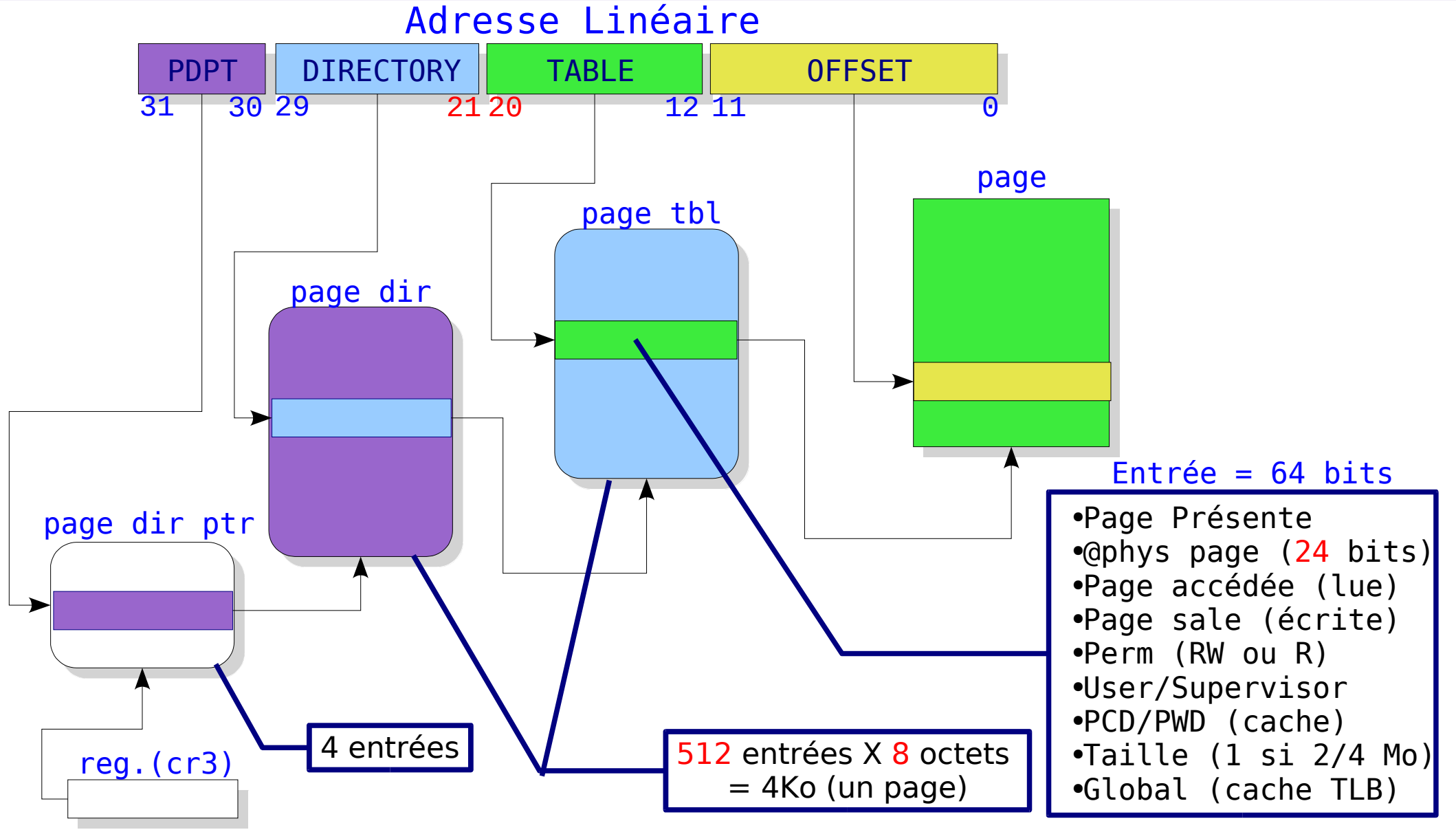
Registres de Pagination du 80x86

- Registres de Contrôle :
  - cr0 :
    - PG=1 active la pagination
    - CD=1 désactive cache matériel
    - NW : séléction stratégie cache (write-thr./write-bck)
  - cr2 : @ linéaire sauvegardée lors d'un défaut de page
  - cr3 : @phys du Répertoire de Pages (Page Directory)
  - cr4 :
    - PSE=1 active la pagination étendue
    - PAE=1 active l'extension d'adressage physique (36 bits)
    - PGE=1 active le mode « Page Global Enable » (évite de vider l'entrée de la page du cache TLB)

La pagination « étendue » des 80x86



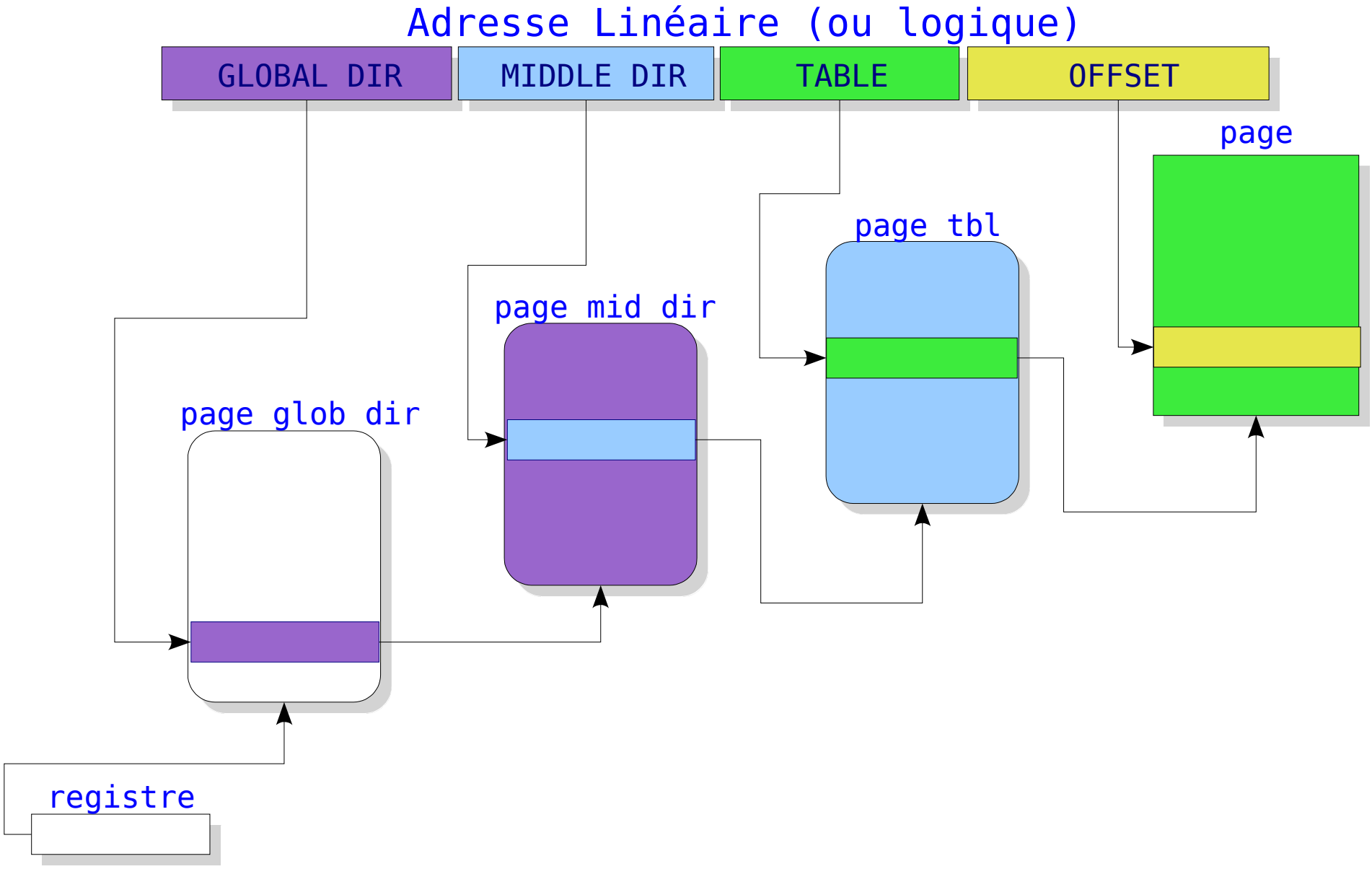
PAE x86 : Extension d'Adr. Physique (64 Go)



La Pagination dans Linux

- Modèle de Pagination à 3 niveaux
  - Inspiré du modèle de pagination des CPU Alpha
    - Page Global Directory
    - Page Middle Directory
    - Page Table
  - Compatibilité facile avec le mode PAE 36 bits des CPU Intel
    - MAIS des limitations : l'adressage linéaire reste sur 32 bits
      - PAE permet d'accéder aux 64 Go via des « fenêtres » de 4Go
  - Compatibilité avec le mode « normal » à 2 niveaux ?
    - Elimination du niveau intermédiaire (PMD) : 0 bits significatifs dans l'@ linéaire
    - Chaînage des tables via une table « sentinelle » à une seule entrée

Modèle Général de Pagination Linux



Organis. de l'Espace d'Adressage Linéaire

- L'Espace d'Adressage Linéaire est limité à 4Go
  - Quelle que soit la quantité de mémoire physique
    - Rappel : jusqu'à 64 Go Possibles !
  - A tout moment un processus ne peut utiliser que 4Go
    - 3 Go pour le mode utilisateur (mémoire virtuelle propre)
    - 1 Go (supplémentaire) en mode système
      - Tous les processus voient la même chose en mode noyau
  - Comment le noyau fait-il pour accéder à la mémoire physique ?
    - Il doit avoir accès à toute la mémoire physique (jsq'à 64Go)
    - En tout et pour tout d'une « fenêtre » de 1Go

Mapping Adressage Linéaire/Physique

- 3 Go utilisateur => adressage virtuel (pagination)
  - Inutilisables par le système
- 1 Go Système
  - 896 Mo (max) : mapping linéaire entre AL et AP
    - Une fenètre de 896 Mo (maxi) sur la mémoire physique
    - Translation d'adresses simplifiée
      - Mapping sur le 4e Go (à partir de `0xc0000000`)
      - **Macros** `__pa(kaddr)` / `__va(phys_addr)`
  - 128 Mo (min) :
    - Allocations non contigues
    - Accès à la mémoire physique haute
      - Fenêtres soit temporaires soit permanentes

Zones de Mémoire Physique

- Linux découpe l'espace mémoire **physique** en 3 zones
  - ZONE_DMA
    - Les bus ISA sont limités à 16 Mo d'espace d'adressage
    - Cette zone contient les adresses physiques (cadres) entre 0 et 16 Mo
  - ZONE_NORMAL
    - Les adresses physiques entre 16 Mo 896 Mo (max)
    - Accès direct depuis l'espace d'AL réservé du noyau
  - ZONE_HIGHMEM
    - Mémoire physique au-delà de 896 Mo
    - Accessible uniquement via la fenêtre de 128 Mo

Organisation des Espaces d'Adressage