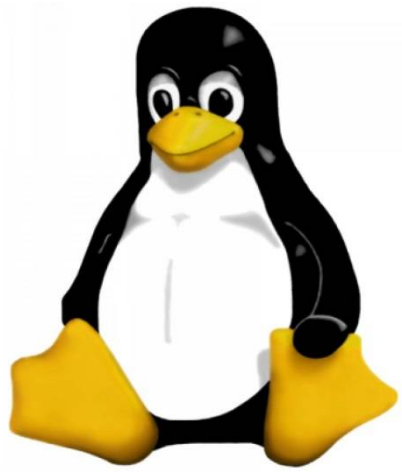
Systèmes d’exploitation

études de cas



Linux

https://www.kernel.org/doc/gorman/pdf/understand.pdf

Complexité du noyau Linux

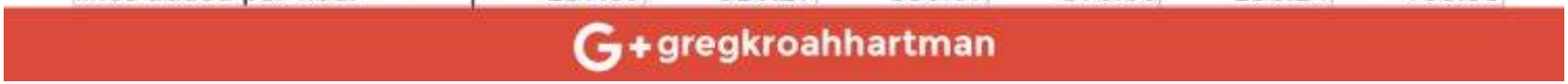
Version	Release Date	Total Size	Size of mm/	Line Count
1.0	March 13, 1992	5.9MiB	96KiB	3,109
1.2.13	February 8, 1995	11MiB	136KiB	4,531
2.0.39	January 9, 2001	35MiB	204KiB	6,792
2.2.22	September 16, 2002	93MiB	292KiB	9,554
2.4.22	August 25, 2003	181MiB	436KiB	15,724
2.6.0-test4	August 22, 2003	261MiB	604KiB	21,714

«Understanding the Linux® Virtual Memory Manager » Mel Gorman. Pearson Education, 2004 (748 pages)
https://pdos.csail.mit.edu/~sbw/links/gorman_book.pdf

Commande système (version de la distribution utilisée) :
lsb_release –a
uname -a

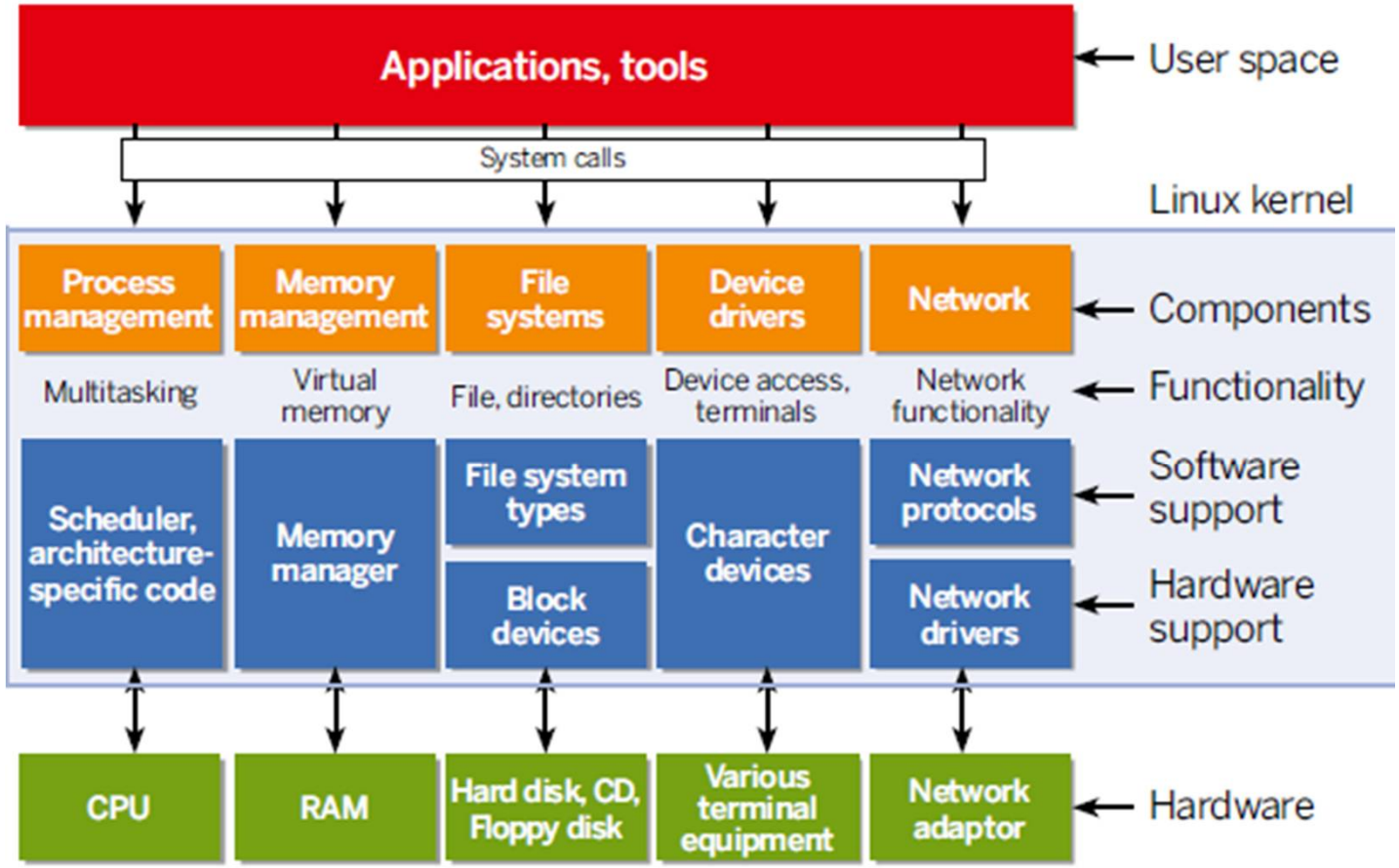
Noyau Linux – versions

version	v4.7	v4.8	v4.9	v4.10	v4.11	v4.12
date released	07/24/16	10/02/16	12/11/16	02/19/17	04/30/17	07/02/17
days	70	70	70	70	70	63
commits	12283	13382	16214	13029	12724	14570
number files	54376	55476	56206	57172	57964	59806
number lines	21720955	22071048	22348356	22839659	23137402	24170860
lines added	399649	548129	554512	632782	430480	1202920
lines removed	100563	198005	275085	136735	132297	168962
lines modified	118494	143757	168129	146030	126771	141550
number developers	1582	1596	1729	1680	1741	1821
number employers	223	219	229	222	219	220
changes per day	175.47	191.17	231.63	186.13	181.77	231.27
changes per hour	7.31	7.97	9.65	7.76	7.57	9.64
% growth files	1.38	2.02	1.32	1.72	1.39	3.18
% growth lines	1.39	1.61	1.26	2.2	1.3	4.47
lines added per day	5709.27	7830.41	7921.6	9039.74	6149.71	19093.97
lines removed per day	1436.61	2828.64	3929.79	1953.36	1889.96	2681.94
lines changed per day	1692.77	2053.67	2401.84	2086.14	1811.01	2246.83
lines added per hour	237.89	326.27	330.07	376.66	256.24	795.58



https://plus.google.com/+gregkroahhartman/posts/WqnwNP3Jqzh?hl=en-UK

Noyau Linux



The subsystems of the Linux kernel.

- implémenté (essentiellement) en C (partie en Assembler)

Mémoire physique

- fichier /proc/meminfo → MemTotal
- commandes : vmstat, free

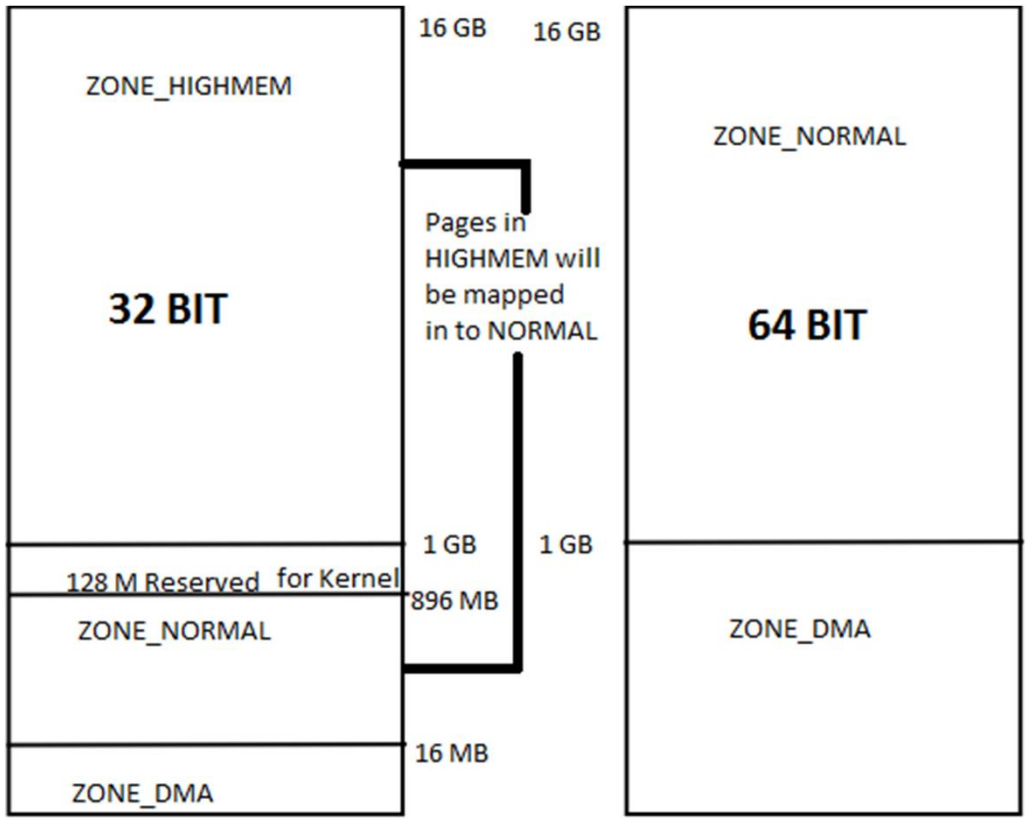
- processeurs x86 : 3 zones

(zone != zone de l'allocation mémoire)

- adressage 32b
  - ZONE_DMA <16MB : (pour des raisons de compatibilité) adressage des premiers 16MB uniquement par des buses ISA
  - **ZONE_NORMAL** 16MB - 896MB :utilisée pour les opérations et allocations du noyau
  - ZONE_HIGHMEM >=896MB : pas d'accès direct du noyau
- fichiers : /proc/pagetypeinfo, /proc/zoneinfo
- divisée en cadres

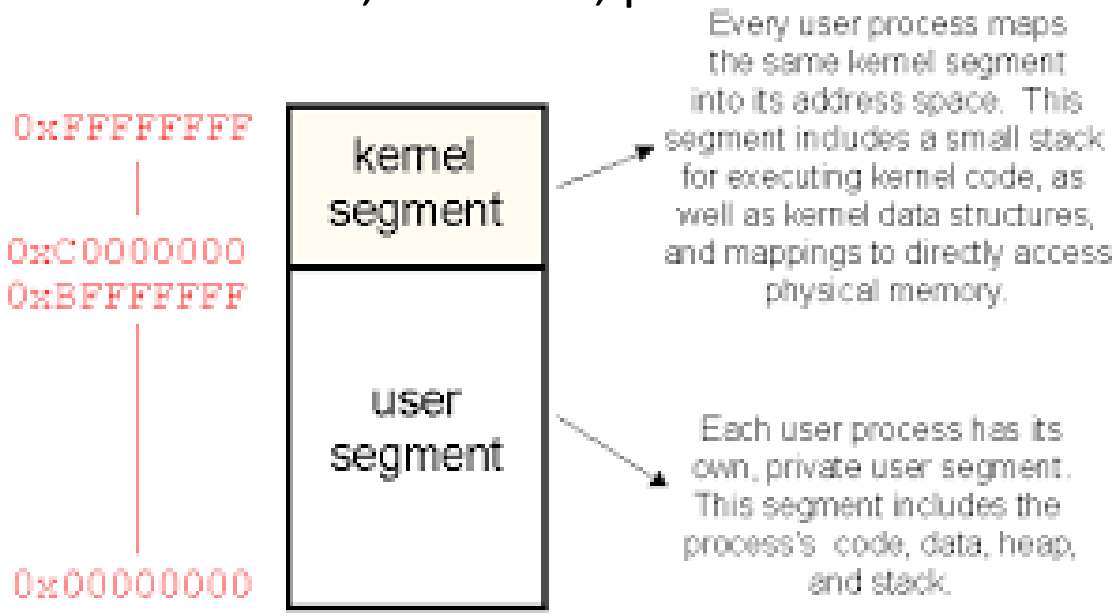
Zones mémoire physique (32b/64b)

exemple de RAM à 16GB



Espace mémoire processus (1)

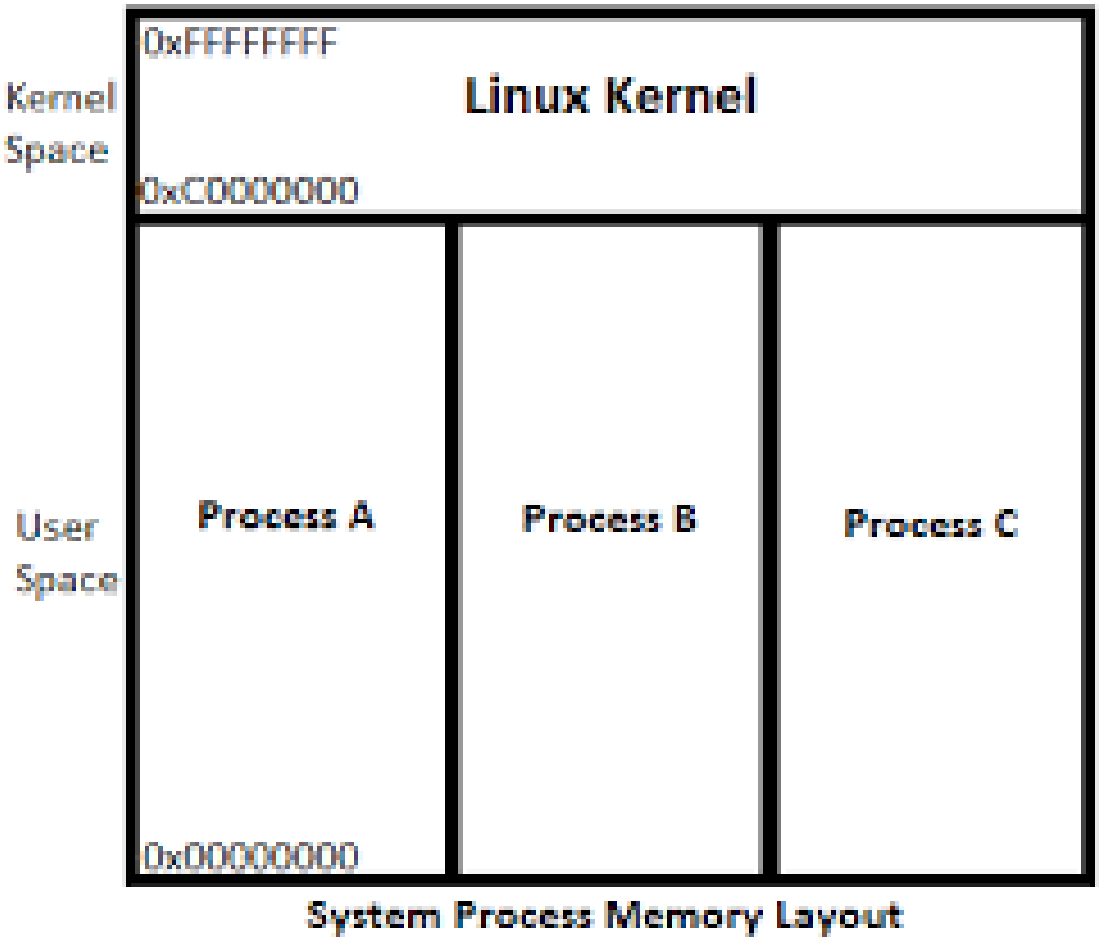
- architecture 32b : taille mémoire processus 4GB
  - 1GB noyau : code et structure de données noyau (adresses identiques dans tous les espaces d'adressage)
  - 3GB utilisateur : code, données, pile



- limite : TASK_SIZE dans include/asm/processor.h

(≡ PAGE_OFFSET = CONFIG_PAGE_OFFSET : paramètre de configuration à la compilation du noyau, fichier sh/boot/Makefile)

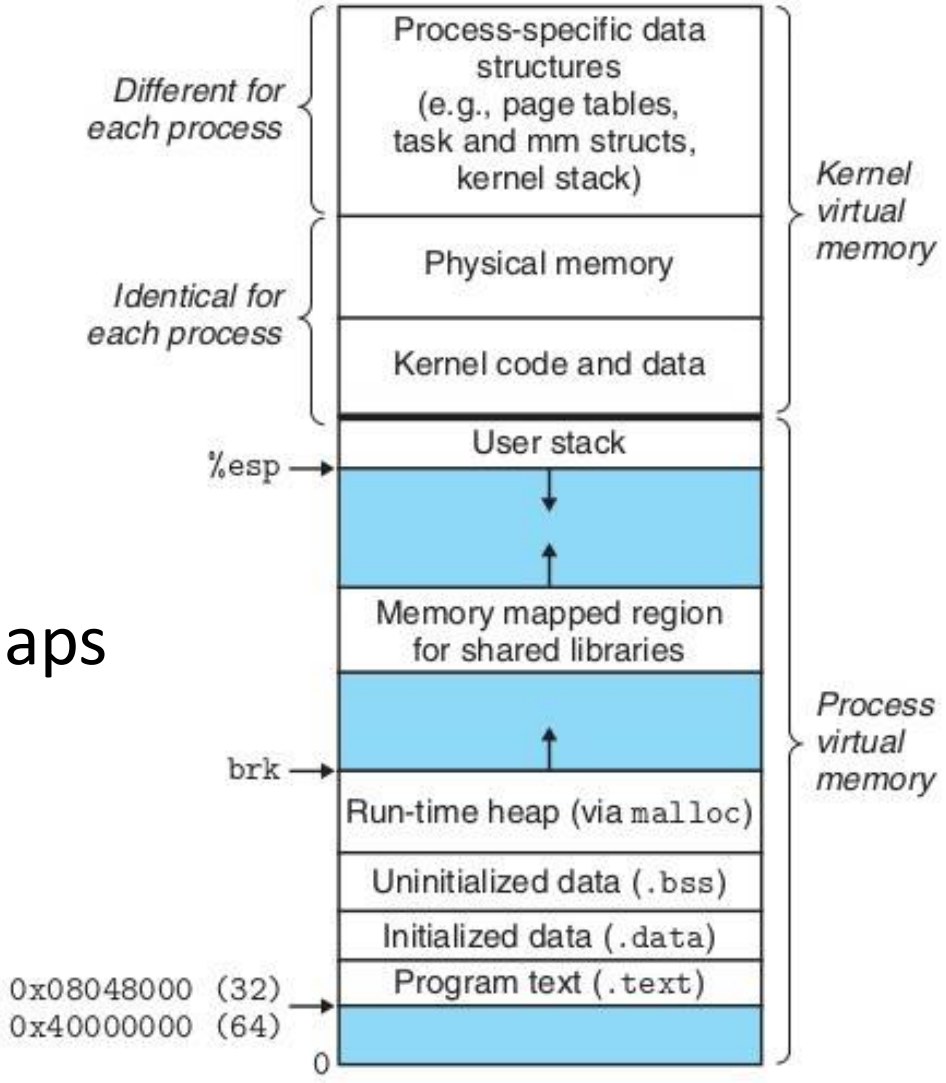
Espace mémoire processus – partie noyau



System Process Memory Layout

Espace mémoire processus (2)

fichier :
/proc/[pid]/maps



Allocation mémoire

- deux approches
  - pagination (pour les processus)
  - mécanismes d'allocation mémoire pour le noyau

Pagination

- mémoire divisée en pages (cadres pour la mémoire physique) de taille 2^p (4KB)
  - taille page

    getconf PAGE_SIZE

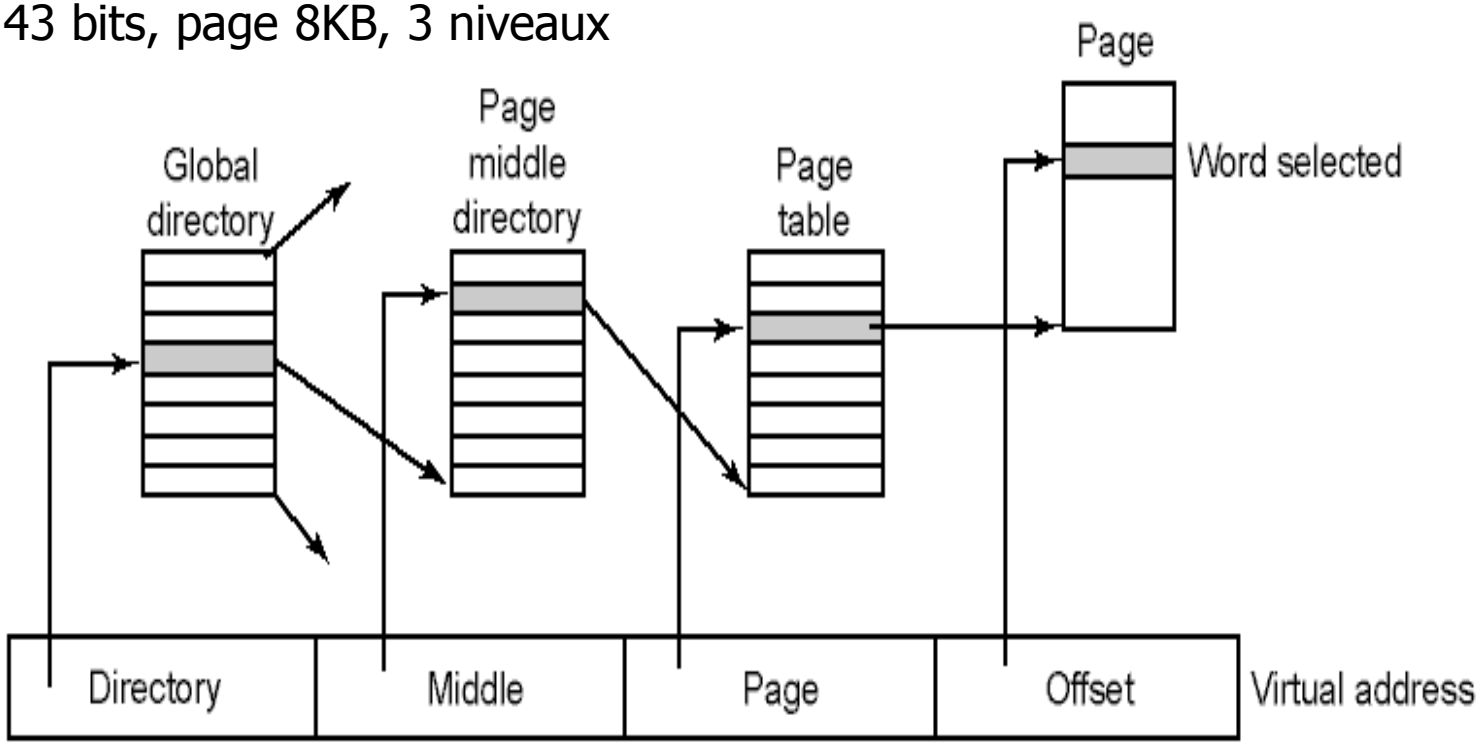
    getconf PAGESIZE

Adressage mémoire virtuelle – Linux

- utilisation réduite des segments (plus de portabilité pour les architectures RISC ne supportant pas la segmentation)
- seulement pour Linux 2.6 avec l'architecture 80×86
- user-mode : programmes utilisent la même paire de segments pour le code et les données (*user code segment*, *user data segment*)
- kernel-mode : même (*kernel code segment*, *kernel data segment*)
- pagination à 3 niveaux (Alpha), 2 niveaux (Pentium)

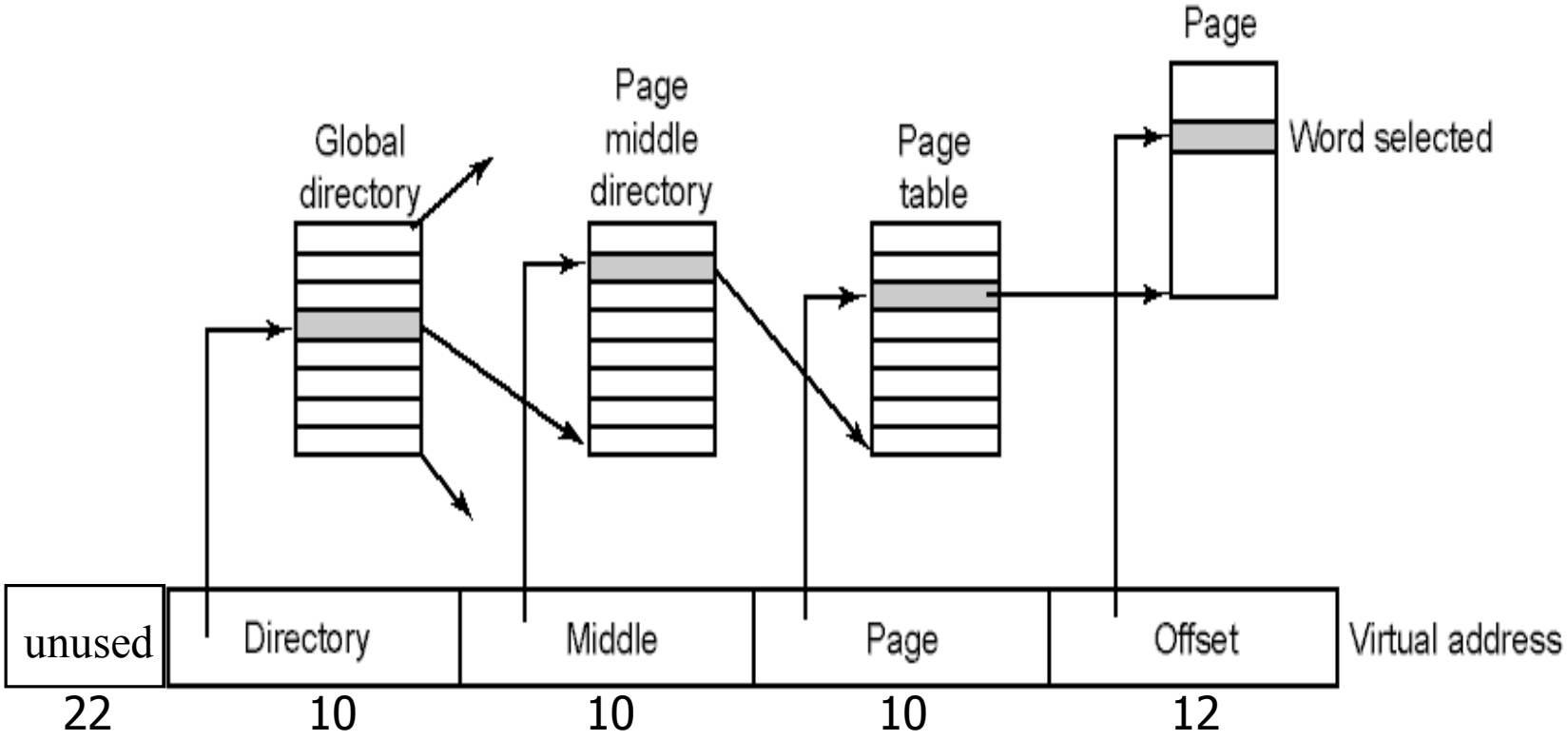
Transcodage – Linux (1)

Architecture Alpha
43 bits, page 8KB, 3 niveaux



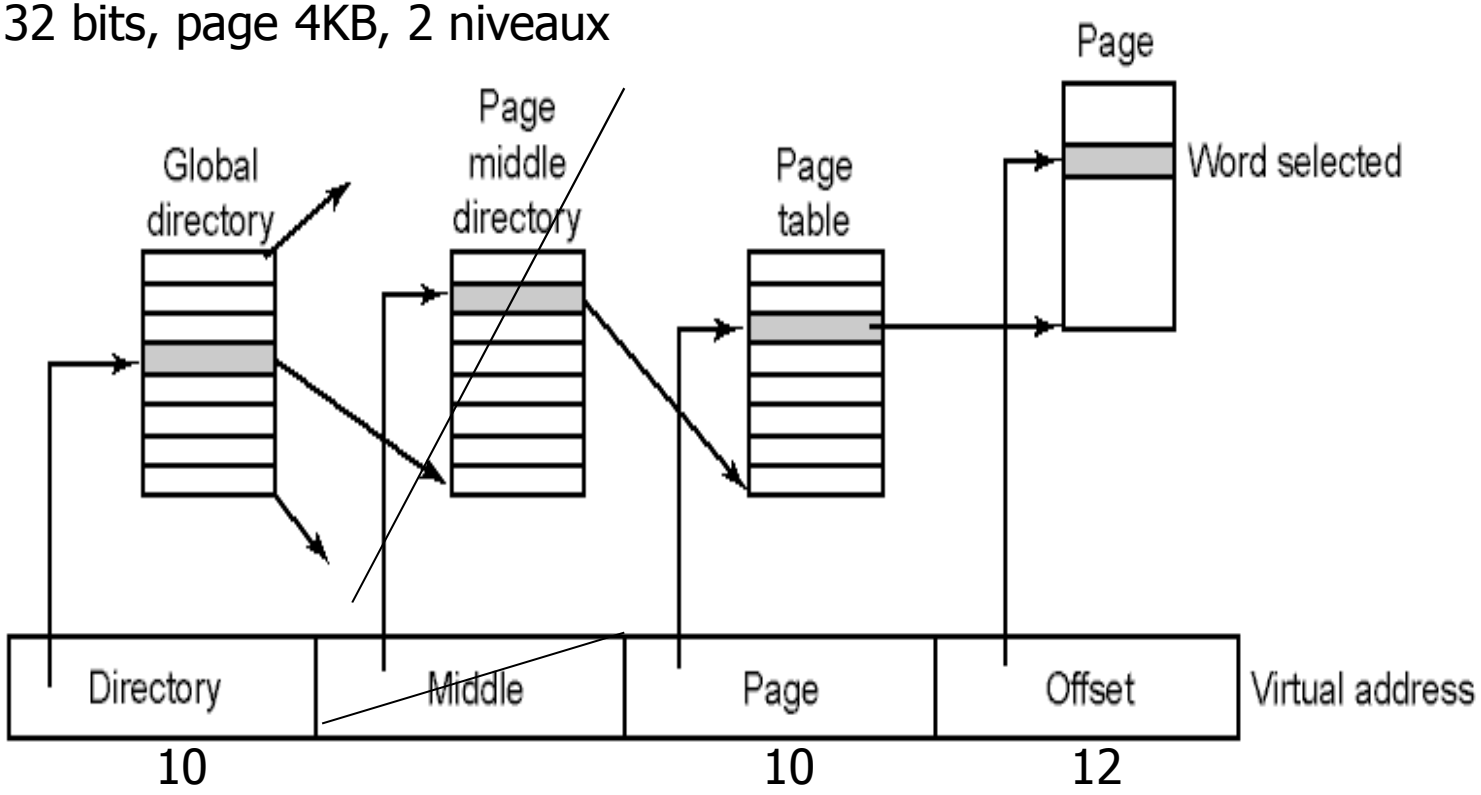
Transcodage – Linux (2)

64 bits, page 4KB, 3 niveaux



Transcodage – Linux (3)

Pentium
32 bits, page 4KB, 2 niveaux



Politique de remplacement de pages

- kswapd (Kernel Swap Daemon)
  - thread noyau responsable de la sélection de pages à retirer de la mémoire centrale
  - activé toutes les 10 secondes
  - il récupère les pages en mémoire centrale si le nombre de cadres libres passe en dessous d'un seuil fixé
- remplacement de pages
  - combinaison de WSClock et LRU-2Q

Allocation mémoire pour le noyau

- mémoire non swappée
- algorithme buddy pour obtenir des espaces contiguës de mémoire
  - les cadres libres groupés en 10 listes de blocs contenant respectivement 1, 2, 4, 8, 16, 32, 64, 128, 256 et 512 cadres libres contiguës
- algorithme SLAB pour des zones mémoires inférieures à la taille d'une page (32 à 4080 octets)
  - pour les structures de données du noyau (descripteurs de fichiers, sémaphores, PCB, etc.)

Ordonnancement Linux

- trois techniques
  - quantum
  - préemption
  - priorité
- classes d'ordonnancement
  - processus temps réel : FIFO et RR (priorité fixe entre 1 et 99)
  - processus ordinaire : temps partagé RR (priorité variable – fct de l'utilisation du processeur)

Linux – ordonnancement FIFO

- FCFS
- ne peut être utilisé qu'avec des priorités fixes supérieures à 0
- le processus le plus prioritaire est celui qui a la plus grande valeur
- un processus s'exécute jusqu'à ce que
  - il soit bloqué par une opération E/S
  - il soit préempté par un processus de priorité supérieure
  - il appelle *sched_yield*

Linux – ordonnancement RR

- amélioration de FIFO
- tranche temporelle (60ms)
- un processus qui a été préempté par un processus de priorité supérieure terminera son quantum de temps lorsqu'il reprendra son exécution

Linux – ordonnancement temps partagé

- ne peut être utilisé qu'avec des priorités fixes égales à 0
- pour les processus qui ne réclament pas de temps réel
- algo CFS (Completely Fair Scheduler) Linux >=2.6.23
  - équité temps d'exécution pour chaque tâche
  - *vruntime* : temps virtuel (estimé) par tâche
  - file d'attente : arbre binaire red-black de tâches par vruntime
- priorité variable (incrémentée à chaque quantum de temps où le processus est prêt mais non sélectionné pour exécution)

Linux – commandes scheduler (1)

- Linux 2.6.32-33-generic #70-Ubuntu SMP

> **chrt** -p 1467
stratégie de planification d'exécution pour pid 1467 actuel : SCHED_OTHER
priorité de planification d'exécution pour le pid 1467 actuel : 0
> chrt -p 3
stratégie de planification d'exécution pour pid 3 actuel : SCHED_FIFO
priorité de planification d'exécution pour le pid 3 actuel : 99

> chrt -m
SCHED_OTHER priorité min/max : 0/0
SCHED_FIFO priorité min/max : 1/99
SCHED_RR priorité min/max : 1/99
SCHED_BATCH priorité min/max : 0/0
SCHED_IDLE priorité min/max : 0/0

Linux – commandes scheduler (2)

```
> less /proc/2290/sched
console-kit-dae (898, #threads: 64)
-------------------------------------------------------
se.vruntime                    :      382590.534248
se.sum_exec_runtime            :      165.334725
se.avg_wakeup                  :       0.814564
se.nr_migrations               :      114
se.nr_wakeups                  :      919
avg_per_cpu                    :       1.450304
nr_switches                    :       1002
nr_voluntary_switches          :       920
nr_involuntary_switches        :       82
se.load.weight                 :       1024
policy                         :       0
prio                           :       120
```

Linux – commandes scheduler (3)

- fichier /proc/sched_debug

Windows

https://msdn.microsoft.com/en-us/library/windows/desktop/aa366525(v=vs.85).aspx (mémoire)

https://docs.microsoft.com/en-us/windows-hardware/drivers/gettingstarted/user-mode-and-kernel-mode (mémoire)

P. Yosifovich, A. Ionescu, M.E. Russinovich, D.A. Solomon « Windows Internals – Part1 – System architecture, processes, threads, memory management and more » Microsoft Press, 2017

Complexité noyau Windows

- Windows 7 (MinWin) : 150 fichiers, 25 MB (sur disque), 40 MB (en mémoire)
- Vista : 5000 fichiers et 4 GB

- langage d'implémentation : C, C++ (parties en C#, Assembler)

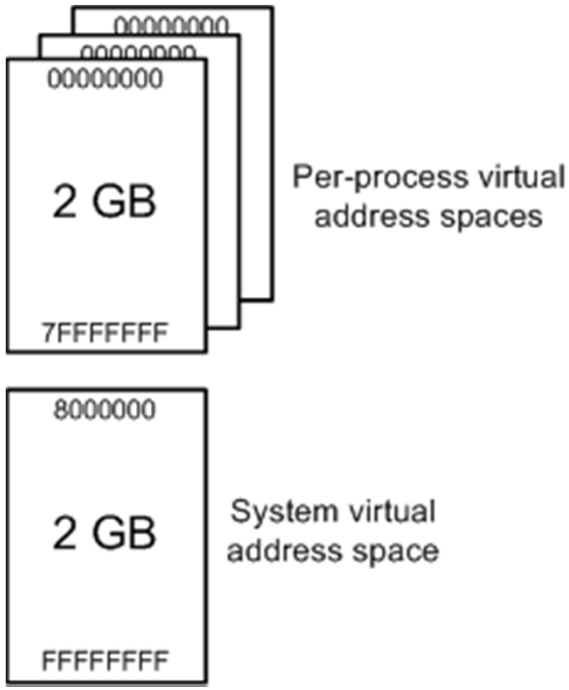
Gestion mémoire

- mémoire physique : de 2GB à 2TB
- mémoire physique : divisée en cadres (=*pages*)

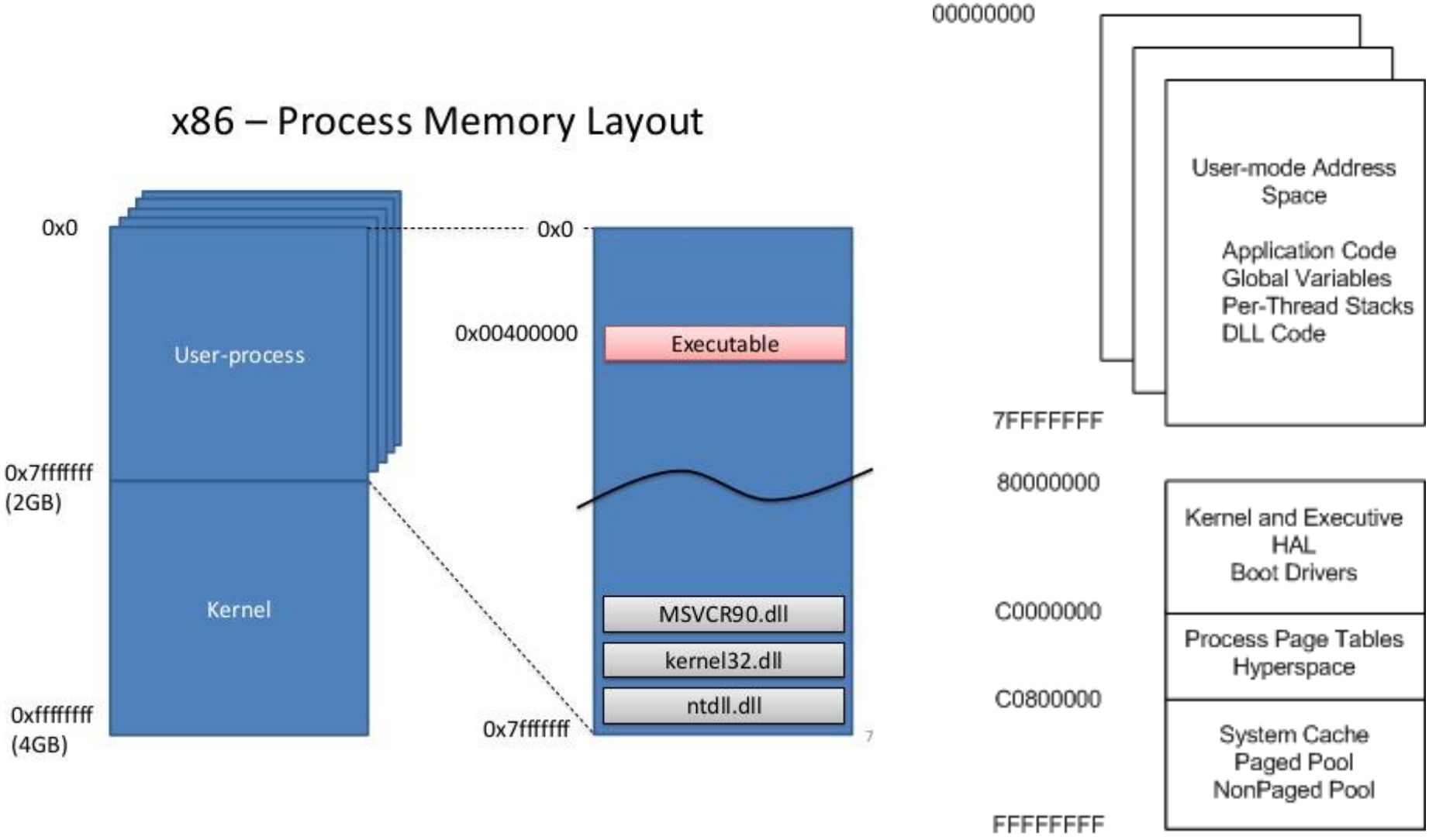
- mémoire virtuelle d'un processus : divisée en *pages*

Mémoire virtuelle processus – 32b

- architecture 32b : 4 GB mémoire
  - 2GB pour le mode utilisateur (de 0x00000000 à 0x7FFFFFFF)
  - 2GB pour le système (de 0x80000000 à 0xFFFFFFFF)

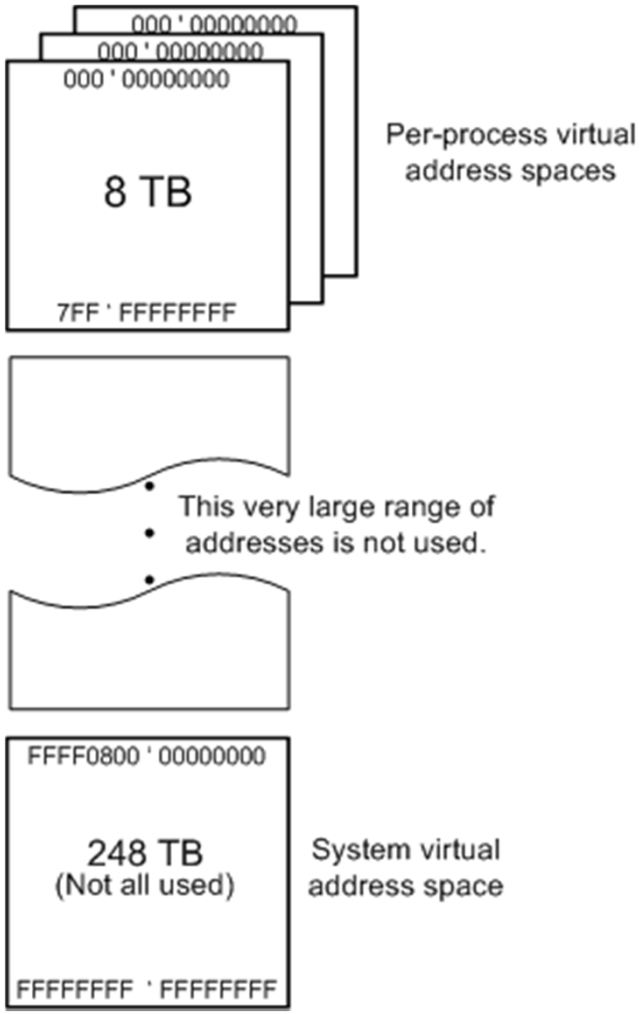


Windows 95, Windows 98, and Windows Millennium Edition

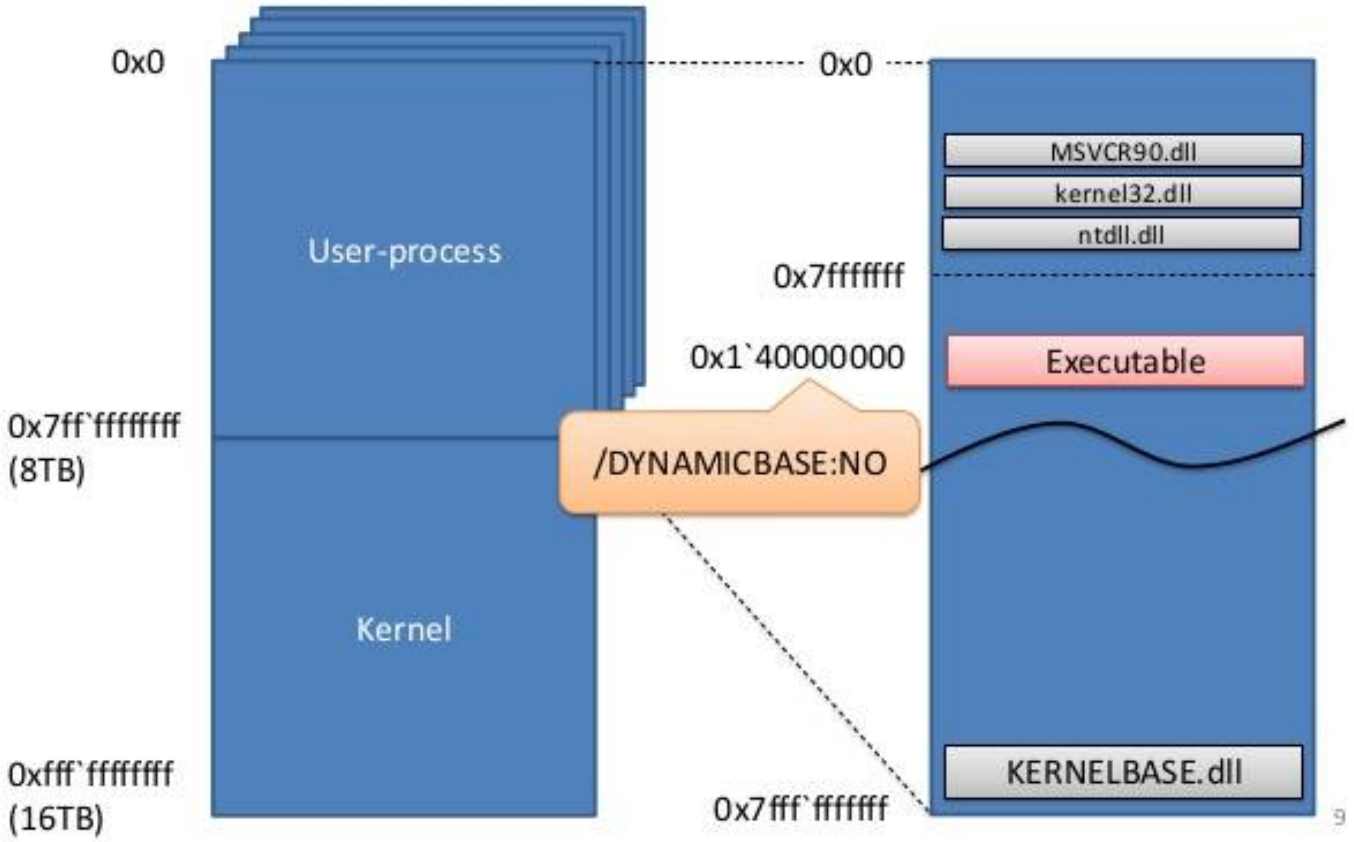


Mémoire virtuelle processus – 64b

- architecture 64b : jusqu'à 2^64o mémoire (=17 179 869 184 GB = 16exaBytes)
  - x64 : 8TB pour le mode utilisateur + 8TB noyau
  - IA64 : 7TB (8-1 réservé pour le *page directory* des adressages WOW64)
    - WOW64 : sous-système Windows capable d'exécuter des applications 32b, incluses dans des versions 64b de Windows



x64 – Process Memory Layout(Cont.)



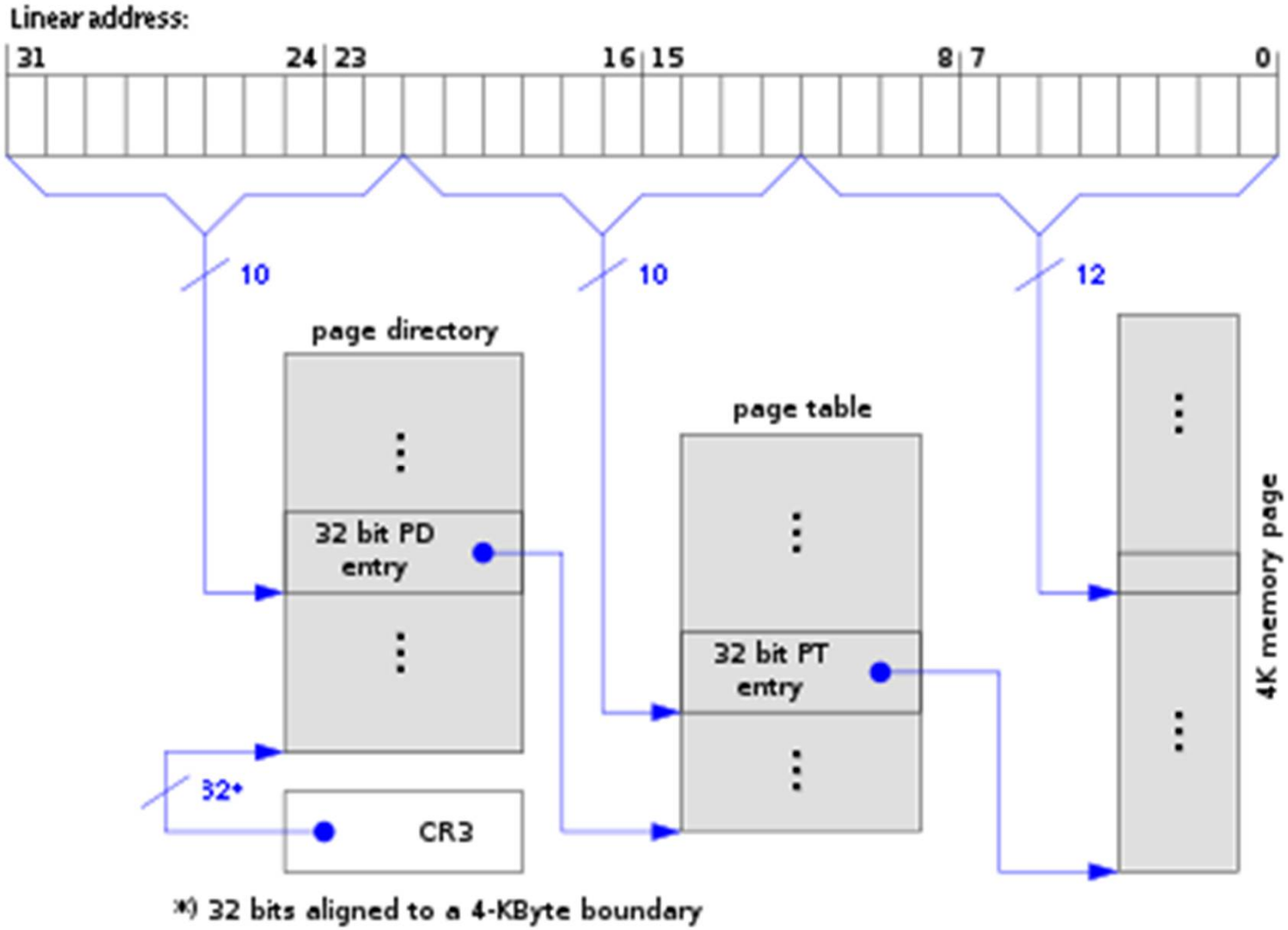
Informations système

- mémoire physique
  - Panneau de configuration $\rightarrow$ Système (mémoire RAM)
- mémoire virtuelle processus
  - gestionnaire de tâches (CTRL+ALT+DEL)
  - *plage de travail* (*Working Set*) : mémoire totale utilisée par un processus (incluant la mémoire privée du processus et la mémoire partagée)

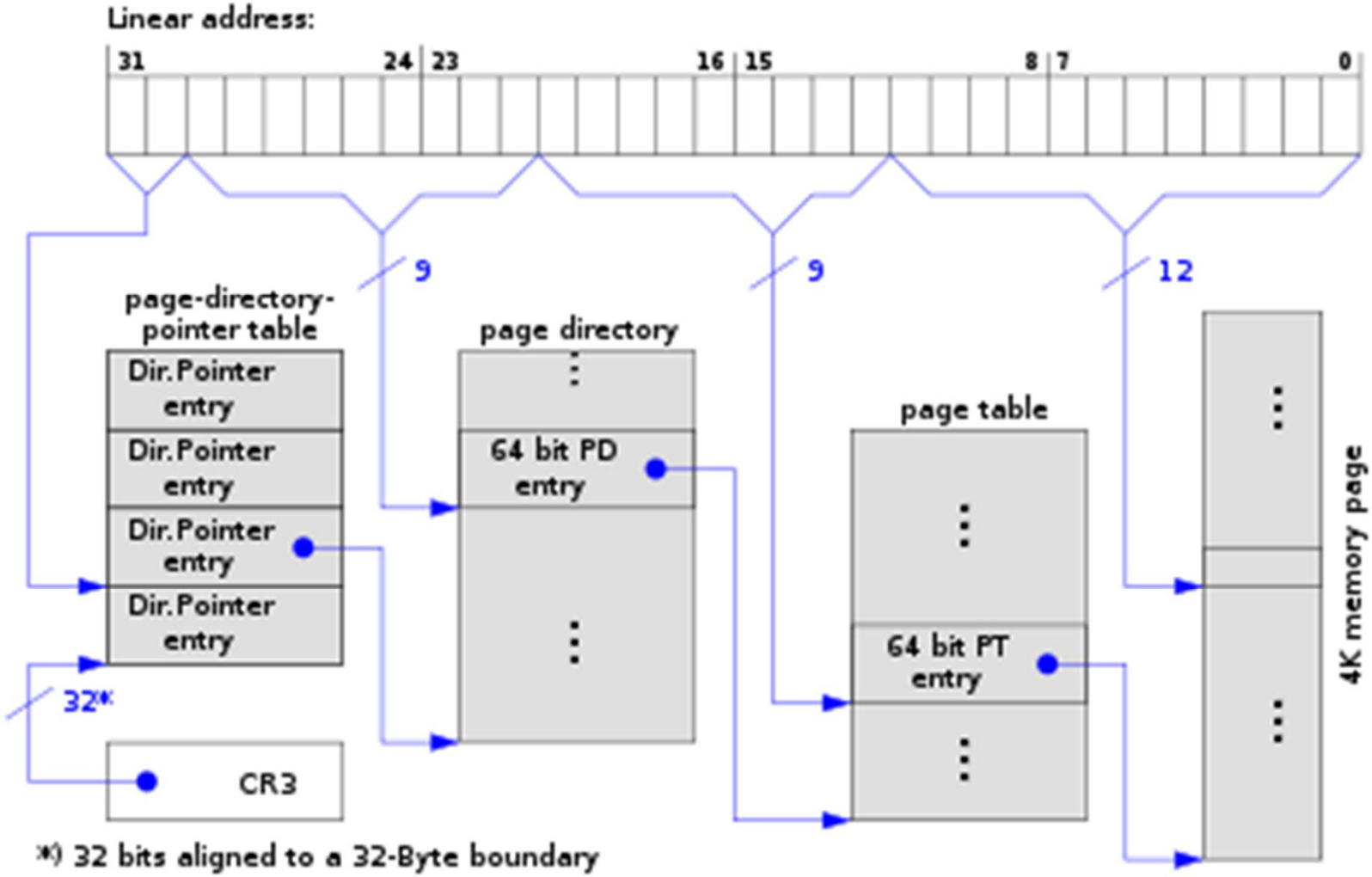
Adressage mémoire virtuelle

- pagination
  - extension PAE (Physical Address Extension) : possibilité pour les processeurs x86 d'accéder plus de 4GB de mémoire physique
  - option PAE disponible : dans le répertoire C:\WINDOWS\system32
    - fichier ntkrnlpa.exe (PAE supportée)
    - fichier ntoskrnl.exe (PAE non supportée)
    - les deux fichiers : option PAE peut être activée/désactivée

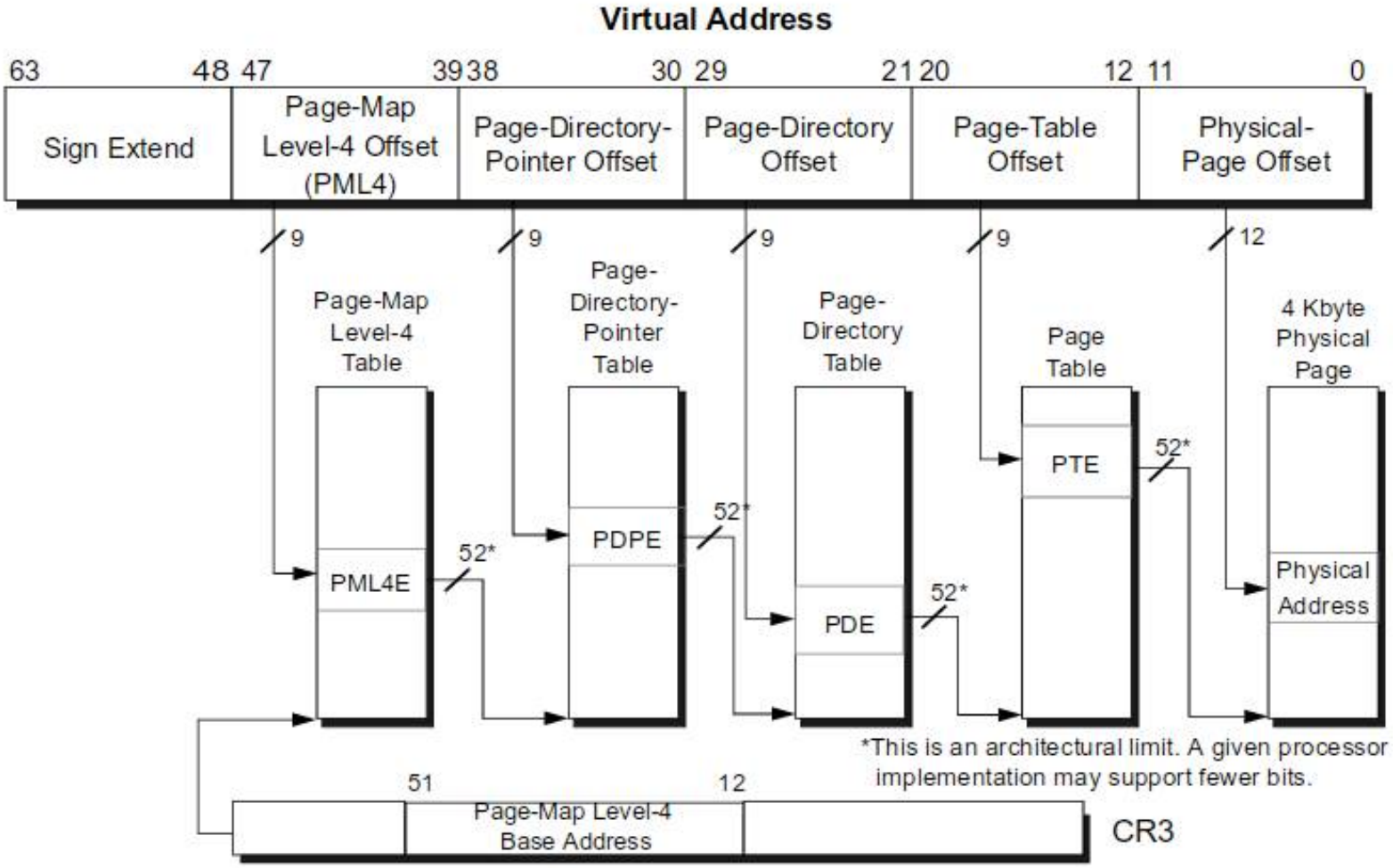
Mode non PAE – 32b



Mode PAE – 32b


Linear address:
31
24
23
16
15
8
7
0
9
9
12
page-directory-pointer table
Dir.Pointer entry
Dir.Pointer entry
Dir.Pointer entry
Dir.Pointer entry
32*
CR3
page directory
64 bit PD entry
page table
64 bit PT entry
4K memory page
*) 32 bits aligned to a 32-Byte boundary

Architecture - 64b

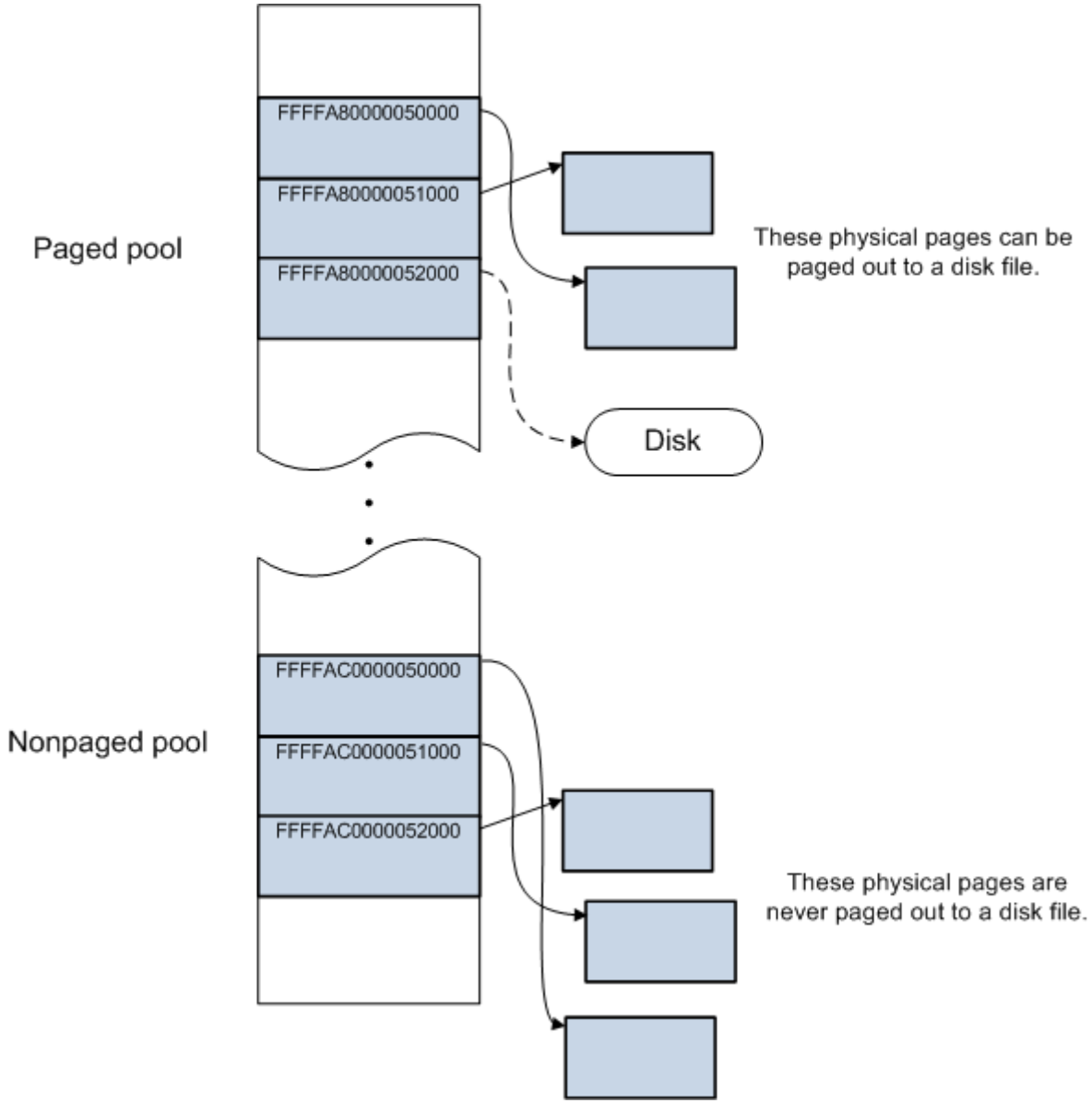


Allocation mémoire

- pagination
- pour les architectures x86 : taille d'une page 4KB

- pagination à la demande + clustering
  - clustering : le défaut de page génère le chargement de la page manquante ainsi que d'un ensemble de pages avoisinantes
  - politique de remplacement de page : algorithme de l'horloge
- paged pool / non paged pool
  - pages de l'espace utilisateur peuvent être déchargées sur disque
  - pages de l'espace système : certaines peuvent être déchargées (paged pool), d'autres non (non paged pool)

Paged pool / non paged pool



Gestion des pages en mémoire

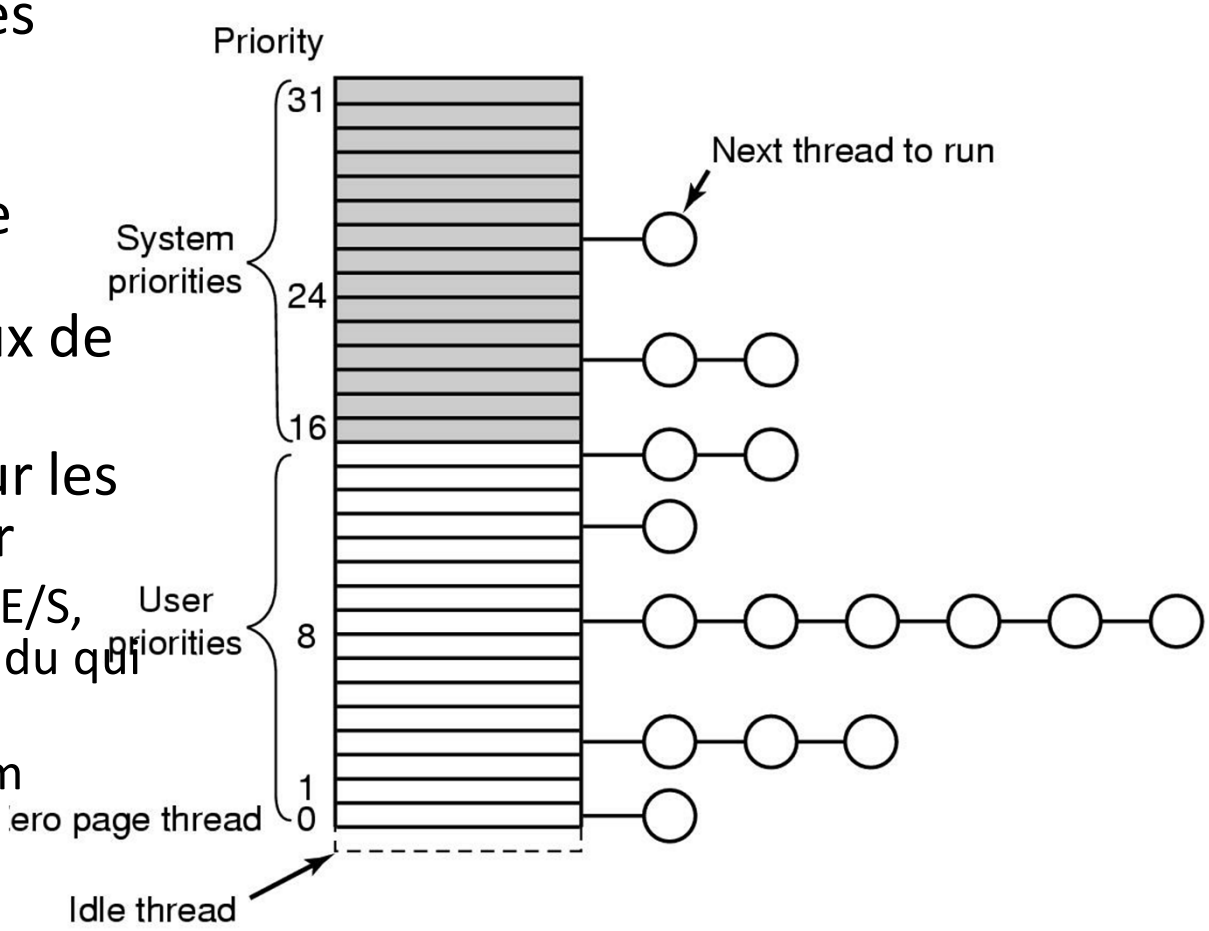
- chaque processus est assigné un ensemble *working set minimum* (20-50) et un ensemble *working set maximum* (45-345)
  - working set minimum : nombre minimum de pages qu'un processus doit être guaranti d'avoir en mémoire
  - working set minimum : nombre maximum de pages qu'un processus peut avoir en mémoire
- *balance set manager* : thread à action régulière (chaque seconde) pour ajuster l'ensemble de pages chargées en mémoire
  - libérer de la mémoire (décharge les pages chargées en mémoire vive) dès que l'espace de mémoire libre passe en dessous d'un seuil
    - victimes : processus dont le nombre de pages chargées dépasse le working set minimum
  - charger des pages en mémoire si le taux de défaut de pages important

Table de pages inversée

- architecture Intel Itanium 64b
  - première solution Windows 64b Server implémente IPT (Inverted Page Table)

Ordonnancement Windows

- ordonnancement des threads (au lieu des processus)
- préemptif, à base de priorité avec files multiples (32 niveaux de priorité) et RR
- priorité variable pour les processus utilisateur
  - augmentée en fin d'E/S, ou évènement attendu qui se produit
  - diminuée si quantum utilisé entièrement



Android

Généralités

- basé sur le noyau Linux 2.6
- conçu pour les dispositifs tactiles (smartphones, tablettes)
- 1 billion de dispositifs Android activés (2011)
- 48 billion d’applications installées (2011)

Machine virtuelle Dalvik

- avant exécution, les applications Android sont converties vers un format compact Dalvik exécutable (.dex) – aux ressources contraintes
- chaque application Android s'exécute dans un processus séparé, instance de la machine virtuelle Dalvik

Gestion mémoire

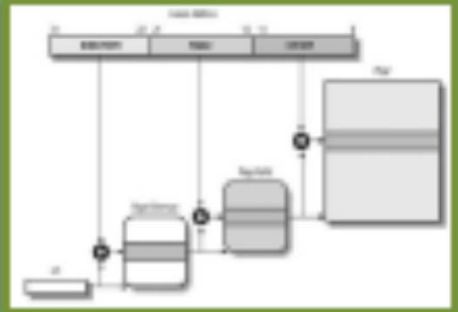
- objectif : minimiser consommation mémoire
- gestion automatique de la mémoire
  - quant une application n'est plus utilisée, le système la suspend en mémoire (pas de ressources utilisées)
  - quand peu de mémoire libre, le système tue des processus qui ont été inactifs depuis un certain temps, dans l'ordre inverse depuis qu'ils ont été utilisés dernièrement (le plus vieux d'abord)

Gestion de la mémoire

Memory Management

- Mobile device: low memory ~ 10 - 20 MB RAM
- *.dex* – Dalvik VM executable file anatomy – smaller size
- *Zygote* –save memory by pre-loading the shared core library classes among applications
- *Ashmem* – the main low-level *mm* addition
- *Garbage Collector* – uses mark-sweep algorithm

- *Paging* and *Fragmentation*
- two-level paging scheme (ARM)
- *External fragmentation*
- *Internal fragmentation*
- *Statistic Analysis* – *mm* directory



Gestion processus

- ordonnancement : RR basé sur la priorité (Linux)
- priorité : background (priorité basse) et foreground (priorité élevée)