Sécurité matérielle des systèmes et des données
Introduction au matériel

Martin Peres

Doctorant de Francine Krief au LaBRI

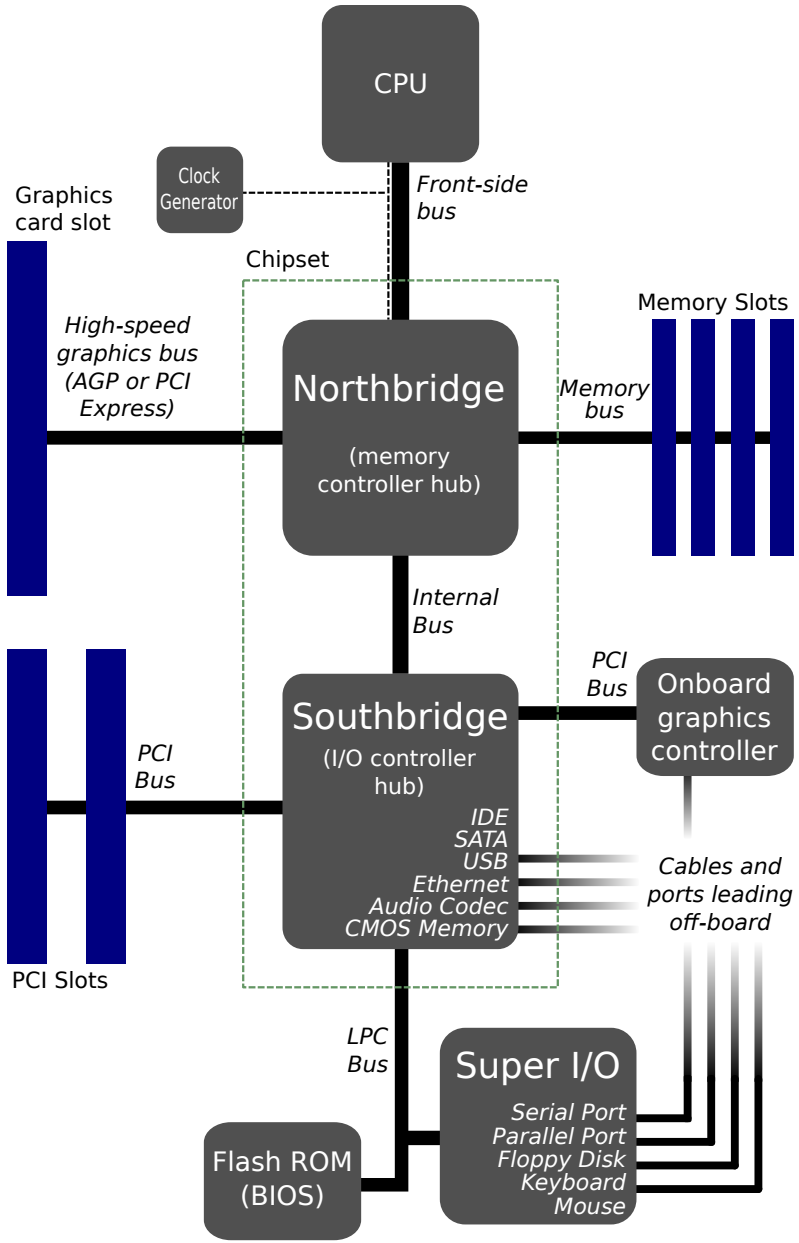
October 14, 2012

Sommaire

L'importance de contrôler les flux d'information

- Segmentation/cloisonnement + contrôle des interactions = Sécurité
- Il donc est important de comprendre les interactions/communications entre les différents périphériques
- La suite du cours vous apportera quelques bases à ce sujet

Comprendre les types d'entrée sortie



Composants importants

- CPU: Traitement
- Northbridge: Contrôle la mémoire
- Southbridge: Contrôle les autres IOs
- BIOS: S'occupe d'initialiser le matériel puis démarre l'OS

Méthodes de transfert entre périphériques

- PIO: Programmed input/output
  - PMIO: Port-mapped input/output
  - MMIO: Memory-mapped input/output
- DMA: Direct memory access
- IRQ(Interruption request) matérielle

PIO

PIO est une entrée/sortie programmée par le CPU à destination d'un périphérique:

- PMIO: Fonctionnement analogue à une socket (read/write). Effectué sur x86 par la famille d'instructions assembleur in[bwl] et out[bwl].
- MMIO: Projection dans l'espace d'adressage du CPU de l'espace mémoire du périphérique puis utilisation des instructions assembleur mov

DMA

Communication directe entre la mémoire centrale et le périphérique. Opérations pour le transfert:

- (optionnel) Le CPU authorise le périphérique à accéder à la zone mémoire concernée en la mappant dans la IOMMU
- Le CPU transmet au périphérique la zone mémoire devant être accédée (base address + length)
- Le périphérique accède à cette zone et y lit et écrit ce qu'il veut

IRQ

Une IRQ permet à un périphérique d'avertir le CPU qu'il doit effectuer une action (généralement urgente).

- Envoi de l'IRQ par un périphérique sur une broche du PIC (Programmable Interrupt Controller)
- Chaque périphérique est associé à une IRQ line qu'il peut être amené à partager avec d'autres
- Lors de la réception d'une IRQ, le processeur stoppe son exécution et sauvegarde son contexte (context switch)
- Maintenant en espace noyau, il consulte son IVT (Interruption Vector Table) qui indique quelle IV appeler
- $--> $ Paradigme différent du polling (Question/réponse versus évènement)

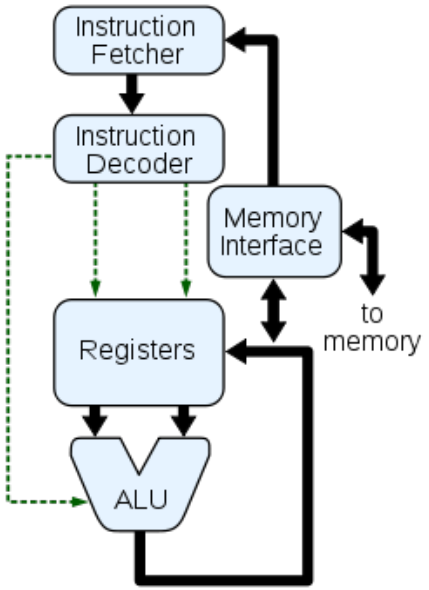
Capacité des différentes interfaces

- Interfaces internes (PCIE, USB Controller, ...): PIO, MMIO, DMA, IRQ
- Interfaces externes: Firewire, Infiniband : DMA (Remote DMA)

Sommaire

Un processeur est défini par

- son architecture:
  - Harvard: 2 bus de données (un pour le code et l'autre pour la data)
  - Von Neumann: un bus pour le code et la data
- sa famille:
  - RISC: Reduced Instruction Set Computer
  - CISC: Complex Instruction Set Computer
- la taille de ses registres (8, 16, 32, 64, 128 bits)
- sa fréquence: De quelques MHz à quelques GHz
- sa finesse de gravure (quelques μm à quelques nm)
- son nombre de coeurs
- la présence de pipelines et leurs types
- ses co-processeurs (FPU, SSE, etc...)



MIPS32 Add Immediate Instruction

001000	00001	00010	0000000101011110
OP Code	Addr 1	Addr 2	Immediate value

Equivalent mnemonic: `addi $r1, $r2, 350`

Les modes d'exécution x86

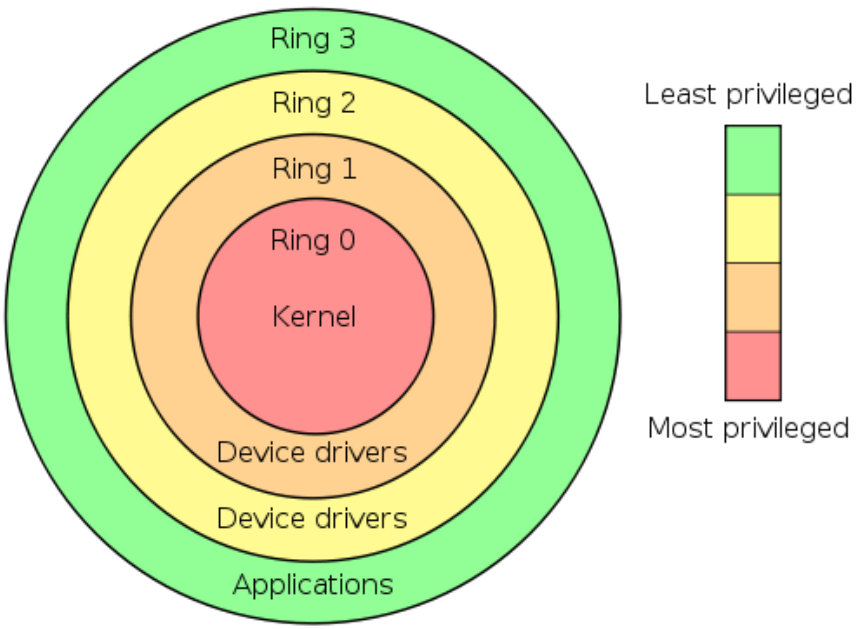
x86 désigne une famille d'architectures developpées par Intel (32 bits). Elle supporte :

- le mode réel:
  - mode 16 bits
  - pas de gestion des droits
  - utilisé au démarrage de la machine
- le mode protégé: mode nominal, 32 bits,
  - mode 32 bits
  - Gestion des droits: MLS 4 niveaux (les rings)
  - mode nominal
- le mode SMM et long (inutile au cours)

Les rings du mode protégé

Le ring actuel défini le current privilège level (CPL). 5 niveaux:

- ring 0: Mode noyau: Accès à (presque) tout le matériel.
- ring 3: Mode utilisateur: Accès limité au matériel.
- ring -1: Mode hyperviseur: Accès aux instructions de virtualisation.



Politique de transition entre les différents niveau d'intégrité

- On ne peut transiter que vers un CPL supérieur
- Retour en ring 0 quand:
  - Interruption logicielle (80h)
  - Interruption matérielle (Timers ou périphériques)
  - Exécution de l'instruction SYSENTER (SYSRET pour revenir au ring 3)

Coût d'un aller-retour entre le ring 3 et le ring 0

- Estimé sur un appel de getpid() sous Linux
- entre 1000 et 1500 cycles processeur (0.6μs sur un processeur à 2.5GHz)
- à comparer avec le coût d'un context switch : 100 cycles

Sommaire

La mémoire vive physique

- Un bit est un mini condensateur (chargé = 1, déchargé = 0)
- Temps de refresh classique, 64ms
- Les bits sont groupés et accédés ensembles (32 bits sur la SDR, 64 bits sur la DDR)
- Peut contenir des codes correcteurs d'erreur (ECC)
- L'adresse de ces groupes est dîte physique

La mémoire vive permet :

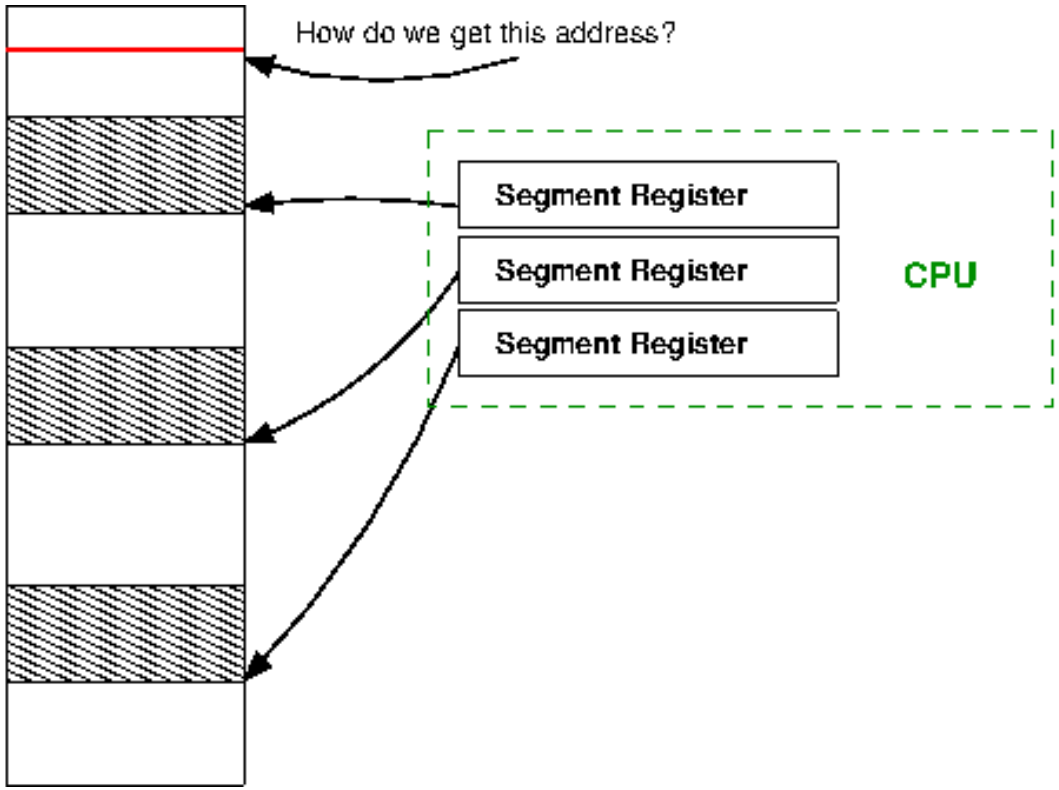
- La lecture/écriture rapide de données de travail

La segmentation mémoire

- Introduite dans les processeurs Intel avec les (80)286
- Découpe la mémoire en segments (base + limit)
- Adresse segmentée = SEGMENT:OFFSET
- Les segments ont :
  - Un type (data, code)
  - Un niveau de privilège (accès si niveau >= CPL)
  - Des restrictions sur les opérations (code: lecture, données: écriture)
- Changement de segment en changeant le registre CS vers le segment voulu de la GDT ou LDT

La segmentation permet :

- de partitionner la mémoire pour cloisonner les applications entres elles



Inconvénient:

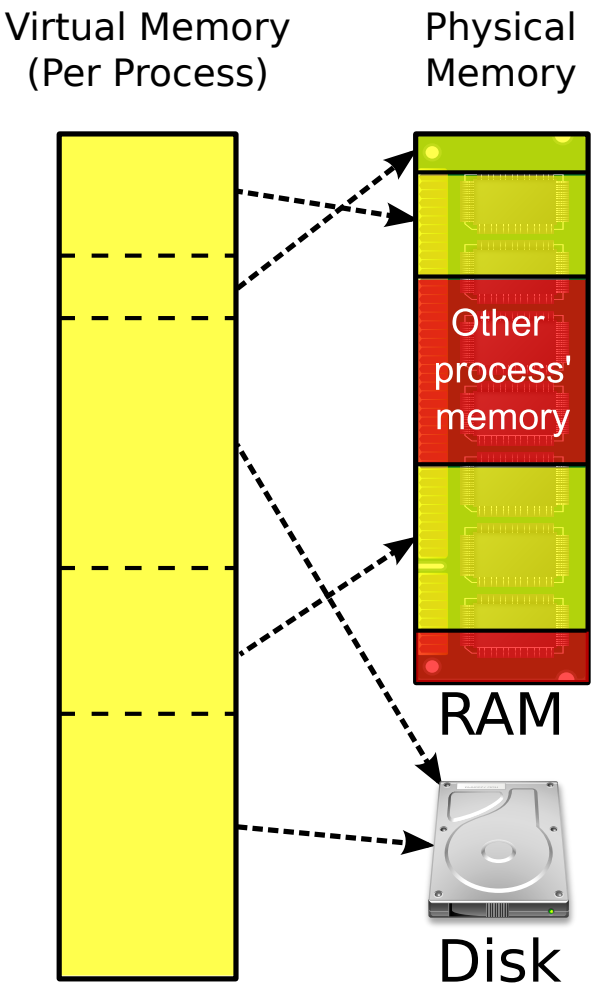
Fragmentation de l'espace d'adressage

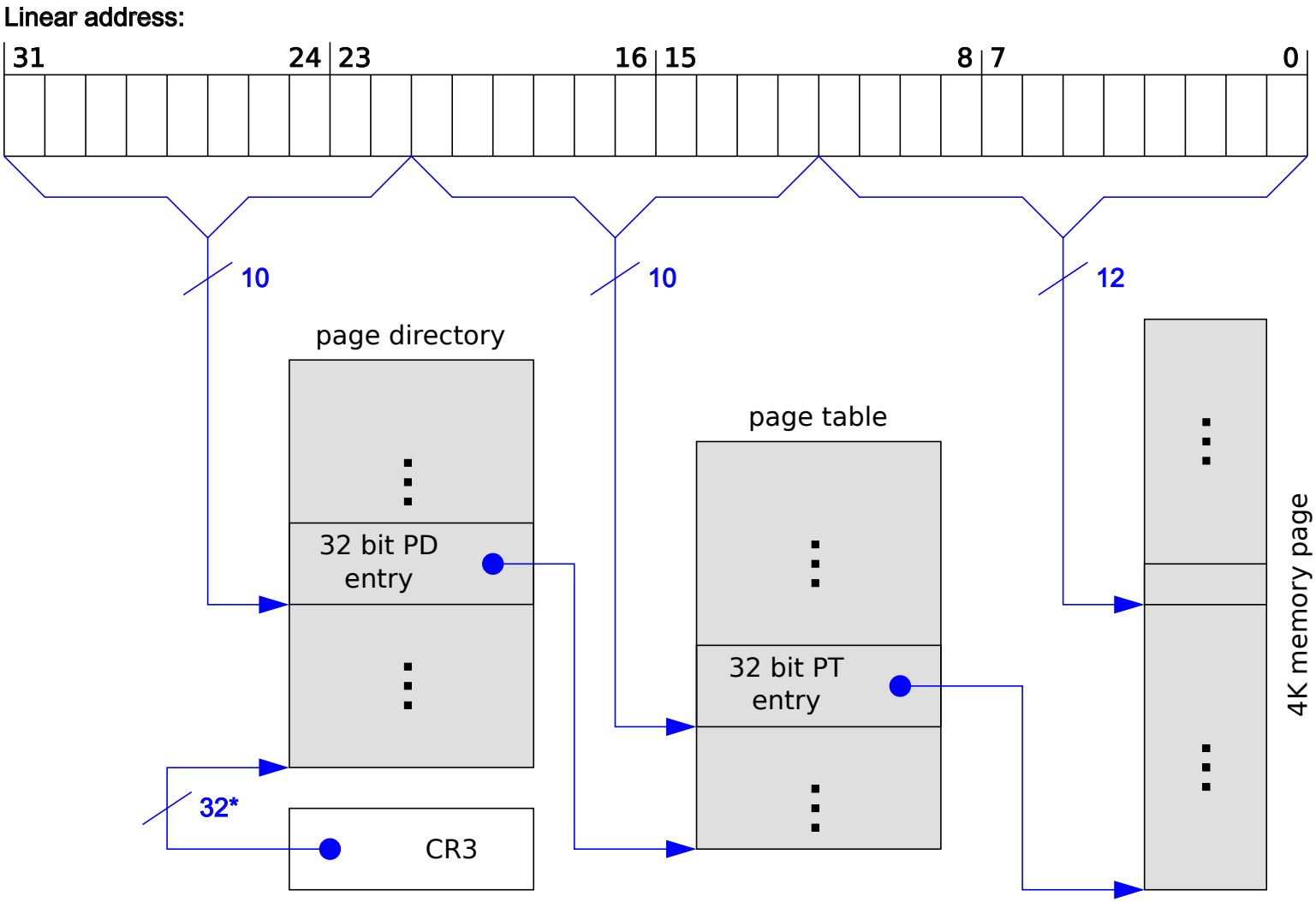
La pagination mémoire

- Introduite dans les processeurs Intel avec les (80)386
- Découpe un espace d'adressage en pages mémoires
- Une page :
  - Taille (généralement) fixe (4kio en général)
  - Permissions d'accès (writable?, kernel/user?)
  - Permission étendue. No eXecute(NX), introduit dans les x86_64

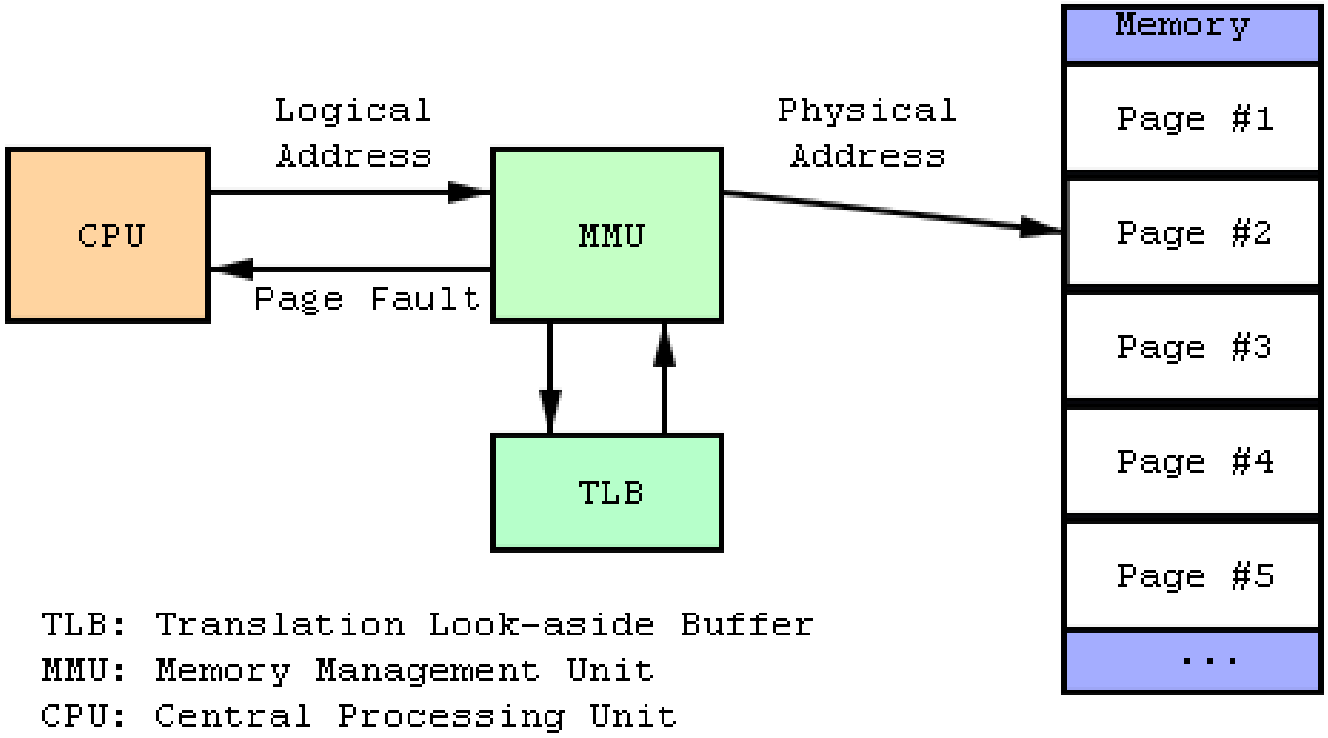
La pagination permet :

- d'avoir un espace d'adressage par processus, en changeant CR3 + flush du TLB, lors d'un context switch (mémoire virtuelle, VM)
- de pouvoir swaper sur le disque dur les pages inutiles
- de ne pas fragmenter la mémoire (vs segmentation)


Virtual Memory
(Per Process)
Physical
Memory
Other
process'
memory
RAM
Disk



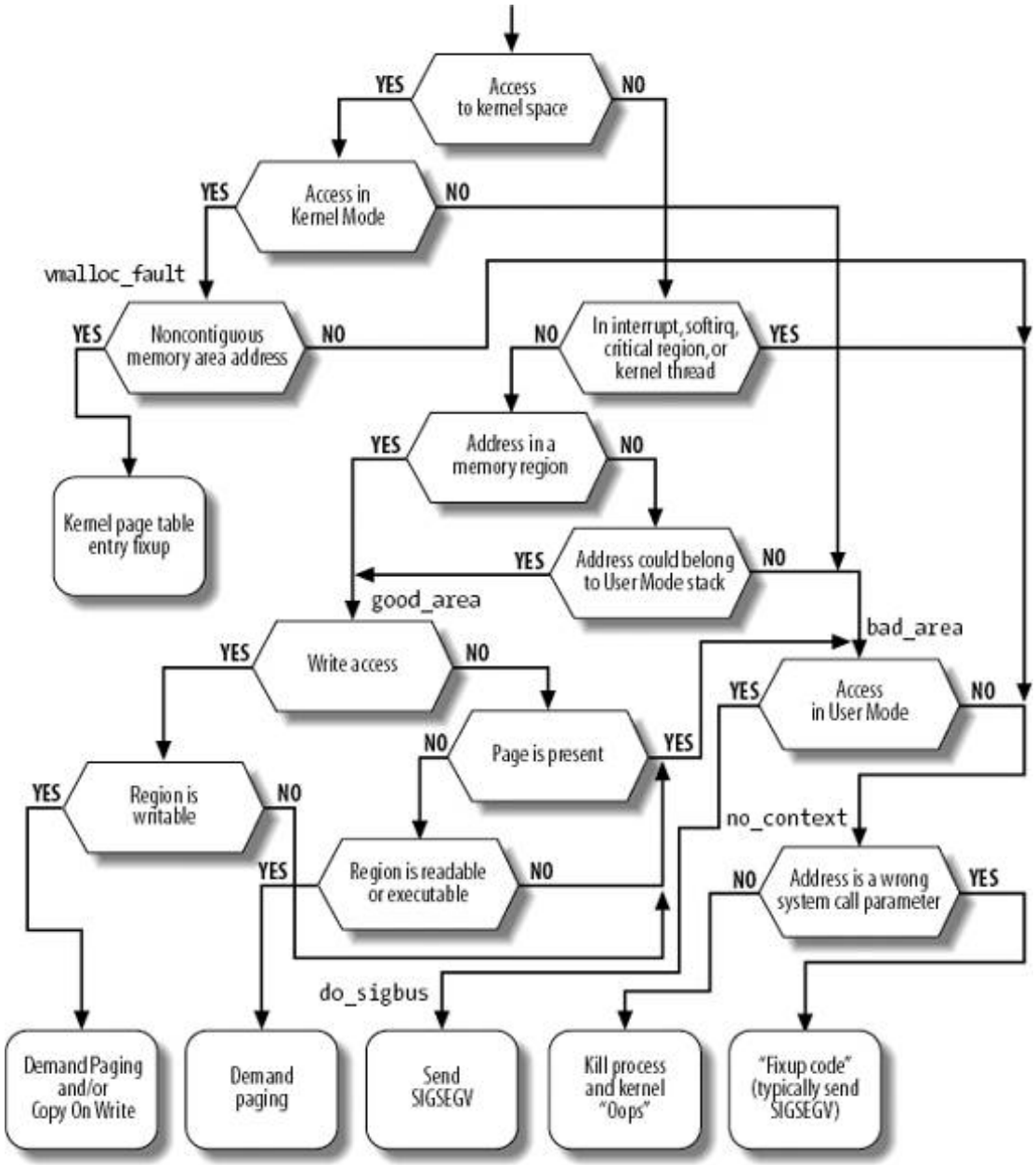
*) 32 bits aligned to a 4-KByte boundary

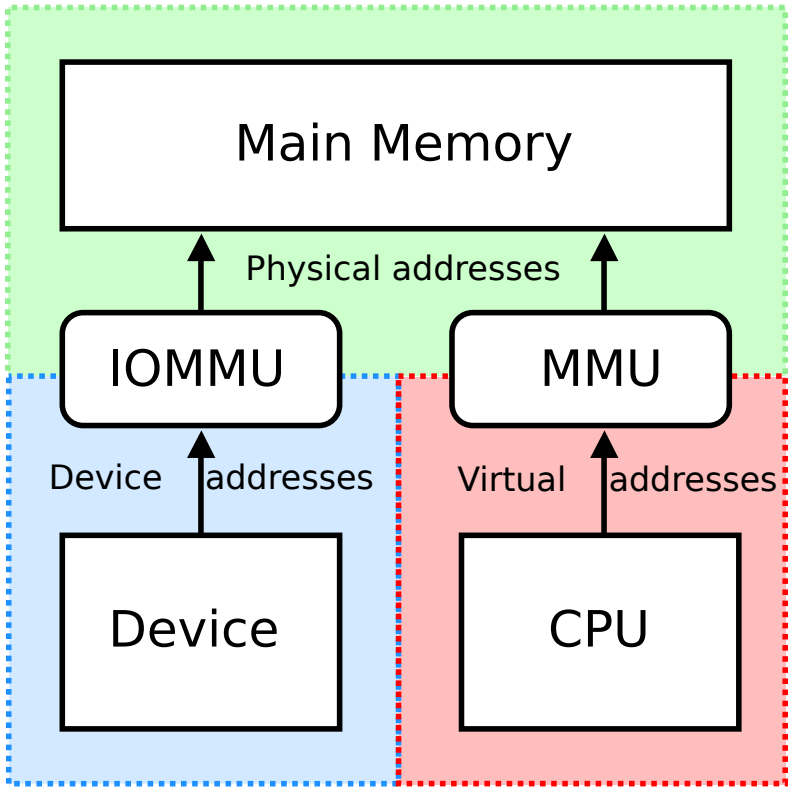


Le TLB

- fait correspondre l'adresse virtuelle et l'adresse physique
- génère un page_fault lorsqu'il ne connaît pas la correspondance
- est vidé à chaque context switch
- est composé de l'iTLB et le dTLB

La pagination mémoire





La IOMMU

- fonctionne de la même façon que la MMU mais pour les périphériques
- n'est généralement pas présente sur un ordinateur

Sommaire

TD

- Listez les menaces s'appliquant au système présenté
- Quelles en sont les vulnérabilités ?
- Quels mécanismes peuvent être utilisés pour bloquer ou mitiger ces vulnérabilités ?

Conclusion

- On peut faire qu'un processus se croit seul sur la machine (Mémoire virtuelle)
- On peut contrôler les flux internes grâce à la IO-MMU
- On peut spécifier des droits d'accès sur les pages et les segments mémoire (rwx, CPL)