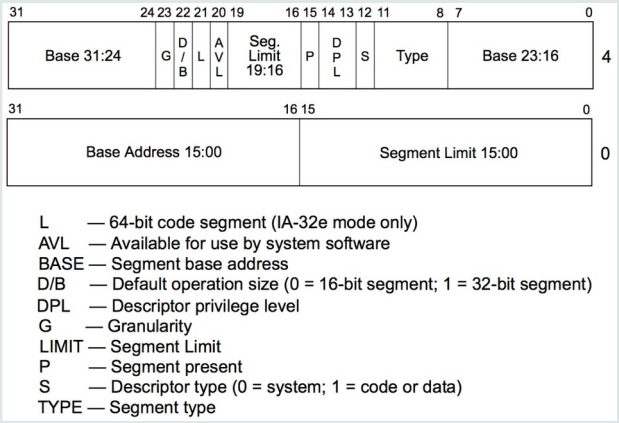
Protection systèmes d'exploitation **TLS-SEC**
Fiche de révision, examen 04/01/2023 10h00 INSA, une feuille A4 papier autorisée

Modes opératoires :
- Mode protégé : mode nominal du processeur, supporte la protection de la mémoire, la ségrégation des instructions par niveaux de privilèges, les rings (0 & 3)
- Mode réel : initial, historique, mode d'exécution 16 bits, adressage 20bits (1 MB), pas protections
- Mode de gestion système (SMM) : le plus privilégié, utilisé par le BIOS, accessible via SMI#
  → Accès sans restrictions aux ressources matérielles, difficilement accessible
- Mode 8086 virtuel : permet au processeur depuis le mode protégé l'exécution de logiciels anciens prévus pour du 8086 (ces logiciels étant exécutés dans un environnement isolé)
- Mode IA-32e (uniq.t sur Intel 64) : sous-mode compatibilité 32bits, équiv. mode protégé mais 32bits
- Mode virtuel (VMX) : gestion virtualisation matérielle, permet de filtrer les instructions, les exceptions, la gestion de la mémoire et les accès aux périphériques
  → 2 modes d'exécution : vmx-root (VMM, pour l'hyperviseur) et vmx-nonroot (VM)

Segmentation mémoire (32 bits, mode protégé)
Espace mémoire : linéaire 32bits (4GB), segmentation possible, adrssge relatif au segmnt (dit logique)
Segmentation : Modèles *flat/protected flat/multi-segment*, segments avec ppts : type, taille, droits
Structures utilisées pour la segmentation (multi-segment en particulier) :
Sélecteurs de segment : utilisé par les registres de segment (CS, SS, DS, ES, etc.), permet d'accéder à un descripteur. Définit le niveau de privilège (bits 0-1), la table (Globale/Locale, bit 2) et un indice (bits 3-15)
Tables de descripteurs de segments :
- GDT : *Global Descriptor Table*, *unique au cœur de processeur (8192 max., premier NULL)*
- LDT : *Local Descriptor Table*, *propre à la tâche*
- IDT : *Interrupt Descriptor Table*, *unique au cœur de processeur*, *pour les interruptions*

Descripteur de segment : Voir ci-contre →
Le *far jump* permet de changer de segment en donnant le sélecteur de segment (*CS*) et l'offset (*EIP*).
Divers : SS - Stack Sgmt, CS - Code Sgmt, DS - Data Sgmt, [E-G]S - Extra Sgmt. *Inutilisée dans les OS modernes*.



Niveaux de privilèges (rings, mode protégé)
Identification du niveau de privilèges :
- CPL, Current Privilege Level, contenu dans CS (niveau de privilège du code en cours)
- RPL, Requestor Privilege Level, au niveau du sélecteur (dans SS, [D-G]S)
- DPL, Descriptor Privilege Level, au niveau du descripteur, doit être supérieur à CPL et RPL

On ne peut charger un descripteur de code qu'à niv. de privilège équiv., changement via call/int gate

Les interruptions et exceptions. **Exception :** produites par des erreurs ou du code, *addr fault → cr2*
Exemple : #GP (erreur, general protection fault), #NP (erreur, seg. not present), #BP (breakpoint)
- Fault, on la gère et on re-exécute l'instruction (#GP, #NP,...)
- Trap, on la gère et on continue après l'instruction (#BP,...)
- Abort, non récupérable

Interruptions. Produites par les périphériques ou du code (int 0x80).
- IRQ, pour les périphériques
- SMI, par le mode gestion système (SMM)
- NMI, par des périphériques souvent liés à l'ACPI
- IPI, entre coeurs d'un CPU, entre CPUs (SMP)

Fonctionnement d'une interruption : L'IDT (localisé avec idtr) contient les descripteurs (segments de code + handler + DPL nécessaire) pour chaque cas.
⇒ Changement de contexte : sauvegarde niv. Priv. + reg. Généraux + registre d'état (EFLAGS)
→ Potentiellement avec changement de niveau de privilège.
→ Utilisation du TSS (Task State Segment) (champ esp0 pour esp)
Appels systèmes
Historiquement : Interrupt Gate avec DPL=3 dans l'IDT → int 0xnn (e.g. int 0x80)
En interne : esp = TSS.esp0 ; push * ; EIP = IDT[0x80].offset + GDT[IDT[0x80].selector].base
De nos jours : *fast system call* (sysenter/sysexit|syscall/sysret)
via configuration de MSR (Model Specific Register) : MSR_SYSENTER_[CS|EIP|ESP]

La pagination (en 32bits):
⇒ L'adresse linéaire n'est plus l'adresse physique en RAM → MMU (*Memory Management Unit*)
⇒ Demand paging (e.g. pile), shm, fork (addr. Virt. eq. mais diff. addr. physique)
Structures : - Un répertoire des pages (page directory, ***PGD***) → uniquement addr. phy. !
- Une ou plusieurs table(s) de pages (page tables, ***PTB***) → uniquement addr. phy. !

→ Addr. PGD dans cr3, chaque table max. 1024 entrées de 4 octets → 4GB d'espace virtuel
PTE : *Page Table Entry* / PDE : *Page Directory Entry* → User (r3) ou Supervisor (r2-0) + info div.
TLB : *Translation Lookaside Buffers* → cache. Flush si changement de cr3 (addr. PGD) (sauf addr global)
Self-mapping : Le PGD s'auto-référence car sinon seule les adresses virtuelles sont connues.
→ Perte de 4MB d'index
Sécurité : Bofbof, bit user/superviseur, bit read/write, historiquement read ⇒ Execute
Amélioration : diff. iTLB / dTLB, Linux PAGEEXEC, bit NX (64 bits)